Advancing computational pathology with deep learning: from patches to gigapixel image-level classification

David Téllez Martín

Advancing computational pathology with deep learning: from patches to gigapixel image-level classification

Typesetting: LaTeX2
Cover design by: David and Miriam Téllez Martín
Printed by: Ridderprint

ISBN: 978-94-6416-521-0

Advancing computational pathology with deep learning: from patches to gigapixel image-level classification

Proefschrift

ter verkrijging van de graad van doctor
aan de Radboud Universiteit Nijmegen
op gezag van de rector magnificus prof. dr. J.H.J.M. van Krieken,
volgens besluit van het college van decanen
in het openbaar te verdedigen op vrijdag 18 juni 2021
om 10.30 uur precies

door

David Téllez Martín

geboren op 10 januari 1991
te Cordoba (Spanje)

Promotoren:	**Dr. J.A.W.M. van der Laak**
	Prof. dr. ir. N. Karssemeijer
Copromotoren:	**Dr. F. Ciompi**
	Dr. G.J.S. Litjens
Manuscriptcommissie:	**Prof. dr. T.M. Heskes**
	Dr. I. Sechopoulos
	Prof. dr. C. Wahlby (Uppsala Universitet, Zweden)

Advancing computational pathology with deep learning: from patches to gigapixel image-level classification

Doctoral Thesis

to obtain the degree of doctor
from Radboud University Nijmegen
on the authority of the Rector Magnificus prof. dr. J.H.J.M. van Krieken,
according to the decision of the Council of Deans
to be defended in public on Friday, June 18, 2021
at 10.30 hours

by

David Téllez Martín

born on January 10, 1991
in Cordoba (Spain)

Supervisors:	**Dr. J.A.W.M. van der Laak**
	Prof. dr. ir. N. Karssemeijer
Co-supervisors:	**Dr. F. Ciompi**
	Dr. G.J.S. Litjens
Doctoral Thesis Committee::	**Prof. dr. T.M. Heskes**
	Dr. I. Sechopoulos
	Prof. dr. C. Wahlby (Uppsala Universitet, Sweden)

dedicado a mi madre y a mi padre

dedicated to my parents

TABLE OF CONTENTS

Introduction

1

1.1 Histopathology

Histopathology is the medical science that studies the microscopic structure of tissue (histology) to investigate and understand human diseases (pathology). In clinical routine, pathologists analyze samples of human tissue under bright-field microscopy in order to determine a diagnosis for the patient[1].

The process starts with tissue removal, a procedure that extracts a piece of human tissue via surgery, biopsy or, less commonly, autopsy. In order to prevent tissue decay and preserve the morphology of the cells for long periods of time, the tissue specimen undergoes fixation, a technique that involves immersing the tissue in formalin. At this point, the fixated tissue block is cut into extremely thin sections of only a few micrometers thick using a microtome. Due to their thin nature, these tissue sections are transparent and delicate, requiring to be mounted on a glass slide and stained before the examination by a pathologist.

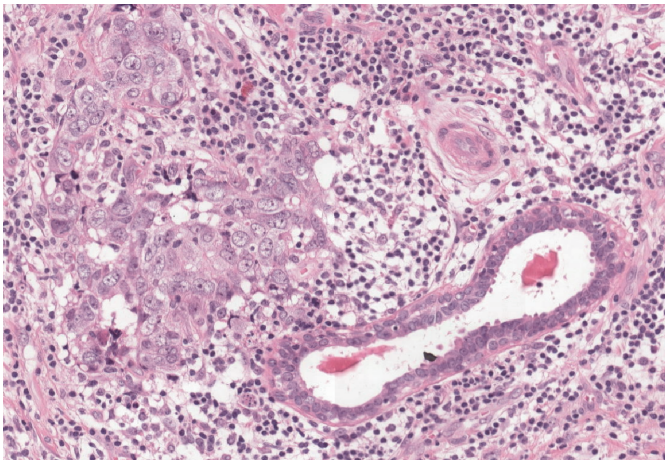

Figure 1.1: Detail of a digitized breast tissue slide stained with hematoxylin and eosin.

The most common staining procedure used worldwide consists of two chemical components: hematoxylin and eosin (often abbreviated as H&E), each one highlighting a distinct morphological feature of the tissue. Hematoxylin stains cell nuclei in blue, while eosin stains the rest (cytoplasm, connective tissue, etc.) in multiple shades of pink (see Fig. 1.1). Pathologists use this stain as the reference for many critical examinations, for example, assessing the tumor grade in breast cancer. Each pathology laboratory uses its own protocol to perform H&E staining, including dif-

ferent chemical concentrations, which results in color deviations across laboratories. It has been shown that even within the same laboratory, differences can be found depending on the day of the week when the staining was performed. This inter-center stain variation is a well-known problem in the field that can even hamper visual inspection by pathologists, and Chapter 3 of this thesis is devoted to studying it.

Depending on the clinical application, a range of staining procedures is available in order to reveal features in tissue sections that are initially hidden to the naked eye. In particular, immunohistochemical staining allows for the detection of specific proteins expressed by the cells contained in the tissue section[2]. This type of staining is more specific than traditional staining procedures like H&E, and therefore may facilitate accurate identification of cells of specific origin (e.g. cytokeratin for epithelial cells[3]), possessing certain characteristics (e.g. estrogen receptor), or undergoing certain transitions (e.g. Ki-67 for proliferating cells[4]). Although immunohistochemistry enhances and complements the morphological analysis performed in H&E, it is usually more resource-intensive and expensive than plain H&E, due to the more specific chemical components and less standardized protocols required to perform it.

1.2 Breast cancer

Breast cancer is the most common type of invasive cancer in women worldwide[5,6], with over a million new diagnoses every year. Early symptoms may include the existence of lumps in the breast, a change in size or appearance of the breast, or the presence of skin abnormalities[7]. After a positive breast examination or as part of a screening protocol, a doctor may prescribe taking a mammogram, i.e., an X-ray image of the breast where internal lesions and abnormalities can be observed. If the radiologist identifies a suspicious lesion in the mammogram, a biopsy is usually performed in order to examine the abnormal cells under the microscope[8].

At this point, the pathologist has to evaluate the biopsy for the presence of cancer, determine the stage and type of cancer, perform the histopathological tumor grading on the tissue specimen, and determine the type of receptors that the potential tumor cells possess. A follow-up surgical resection removes the tumor and verifies the initial analysis. Assessment of the stage of the tumor yields the most important distinction between carcinoma in situ and invasive breast cancer. Furthermore, different types of tumor are recognized, with ductal and lobular being the most common ones and accounting for some 90% of all breast cancers[9]. The primary differ-

ence between in-situ and invasive types is that in the case of in-situ, the tumor cells are contained within certain structures (e.g. milk ducts) and have not invaded tissue outside of them; whereas in the invasive types, tumor cells have broken those barriers and are invading other areas of the breast. Invasive types have a worse prognosis than non-invasive ones due to the tumor cells' ability to invade surrounding tissue, and eventually even spread through the body and give rise to metastases.

Tumor grading is a strong prognostic biomarker for the survival of the patient, since it describes the degree of aggressiveness of the tumor, i.e., how likely the tumor is going to proliferate and grow in the future[10]. Breast cancer grading has three components: (1) nuclear pleomorphism, (2) tubule formation and (3) mitotic count; and can take a value within the 1-3 range, where 3 corresponds to the worst patient prognosis. Nuclear pleomorphism and tubule formation are difficult visual features to quantify objectively, as their analysis goes beyond cell detection and counting. Mitotic count consists of quantifying the number of tumor cells undergoing division in a certain area of the tissue section. It has been shown to be a reliable and independent prognostic marker on its own, although the inter-observer variability of manual counting currently limits its reproducibility[11,12]. In order to develop a more robust and reproducible mitosis counting method, we have devoted Chapter 2 of this thesis to investigate a computer-based solution to automate mitosis counting.

Determining the receptor status of the breast cancer specimen is of utmost importance for the future therapy choice of the patient. Tumor cells might have one of the following receptors: estrogen receptor (ER), progesterone receptor (PR), or human epidermal growth factor receptor 2 (HER2). For example, tumor cells that have an overexpression of ER, i.e., they depend on estrogen to stimulate their growth, can be suppressed by drugs that block estrogen (known as targeted or hormone therapy). When these tumor cells fail to express any of the previous receptors, they are identified as triple-negative breast cancer (TNBC), a variety of breast cancer that cannot benefit from targeted therapy and presents the poorest patient prognosis[13]. In part of this thesis, we focus on studying TNBC patients in an attempt to improve their diagnosis and prognosis.

1.3 Computational pathology

With the advent of computer science and information technology in healthcare, the field of histopathology is undergoing a major digital revolution. In analog pathol-

ogy, doctors examine tissue sections by placing a glass slide under a physical microscope. However in digital pathology, the same task can be completed using an ordinary computer and a screen; and whole-slide scanning is the technology that enables this digitization (see Fig. 1.2). These devices can scan a tissue section and produce a high-resolution digital image of the slide, usually containing more than a billion pixels (gigapixel scale)[14].

Figure 1.2: A whole-slide image scanner accompanied by an attached monitor screen for displaying the digitized slides. Image credit from Leica Biosystems.

These images, commonly known as whole-slide images (WSIs), typically consist of hundreds of thousands of cells and are examined by pathologists using a WSI viewer. This piece of software allows doctors to navigate through the entire tissue sample, providing support to perform basic tasks like zooming, panning, measuring and taking notes[15]. In digital pathology, doctors can benefit from a number of features that are not available in the analog version, e.g., easily sharing cases with colleagues, and creating precise annotations of anomalies and interesting phenomena.

The possibility of having precise annotations depicting abnormal tissue is what has enabled artificial intelligence to be applied to histopathology, a novel field known as Computational Pathology. Computer algorithms can use the information contained in these precise human-made annotations to learn to recognize visual patterns that may be responsible for the patient's disease. In recent years, successful artificial intelligence based methods have been developed and deployed in the clinic, aiding pathologists in certain tasks such as the detection of malignant cells, or the classification and segmentation of a variety of tissue structures[16].

Producing precise manual annotations is an expensive and time-consuming endeav-

our, often requiring the intervention of expert pathologists. However, these annotations are not the only source of information that computer algorithms can exploit in order to learn the causes of a disease. Often, information about the patient is available at almost no cost in the hospital database, e.g., overall patient survival or the result of genetic tests. These labels are commonly known as image-level or patient-level targets, since they are related to the entire WSI instead of being delineated at pixel-level with a precise annotation. An advantage of these image-level labels is that datasets and patient cohorts can scale-up in size very rapidly since they only require scanning the glass slides and performing a database query. On the other hand, specific computer algorithms are required to exploit these labels, which limits the applications that can benefit from them. Exploiting the relationship between image-level labels and WSIs is an active area of research, with Chapters 4 and 5 of this thesis devoted to this idea.

1.4 Deep learning

Deep learning is a field of machine learning that involves training generally large artificial neural networks to solve a task (see Fig. 1.3). Although neural networks have existed and have been studied for many decades[17], they have only recently become mainstream due to their unprecedented success across multiple fields and applications[18]. Recent advances in neural network architectures and training methodologies, the creation of massively large datasets, and the development of specific hardware technology, have all enabled the so-called deep learning revolution[19]. In 2012, Alex Krizhevsky notoriously won the large-scale ImageNet competition by a substantial margin using a type of deep learning based model called convolutional neural network (CNN)[20]. Since then, many researchers have reported drastic improvements in other areas of computer science as well, including computer vision[21], audio processing[22], human speech[23], text generation[24], graphs[25], reinforcement learning[26], generative models[27], and medical imaging[28]. Very recently, worldwide recognized researchers Yoshua Bengio, Geoffrey Hinton and Yann LeCun were awarded the Turing Award for pioneering and laying the foundation for the modern practice of deep learning, and making it a fundamental part of computer science.

Neural networks are at the core of deep learning. These are mathematical models inspired by human neurons in the brain, that perform signal processing, i.e., they transform an input signal into an output signal. For example, converting a picture into a binary response indicating the presence of a certain object in the input image,

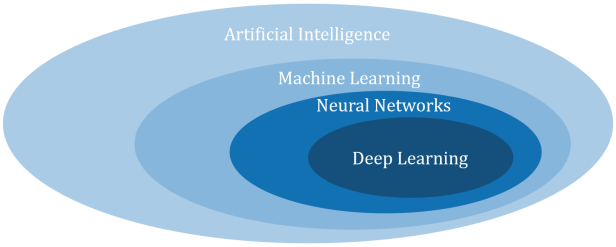

Figure 1.3: Venn diagram depicting the context of Deep Learning.

or translating a sentence in a given language into a foreign one. The most basic case of a neural network is the single-layer perceptron, a type of feedforward neural network whose output is a weighted sum of the input signals followed by a non-linear function [29] (see Fig. 1.4). The weights used to compose the output signal are also known as the parameters of the neural network, and they are automatically learned from the data using optimization techniques. These operations with learned weights are grouped in what is known as a layer, and multiple of these layers are typically stacked sequentially creating very large neural networks. Increasing the number of trainable layers expands the neural network's capacity to learn complex functions, therefore, *deep* neural networks are generally a more suitable choice to solve challenging real-world tasks, hence the *deep learning* term.

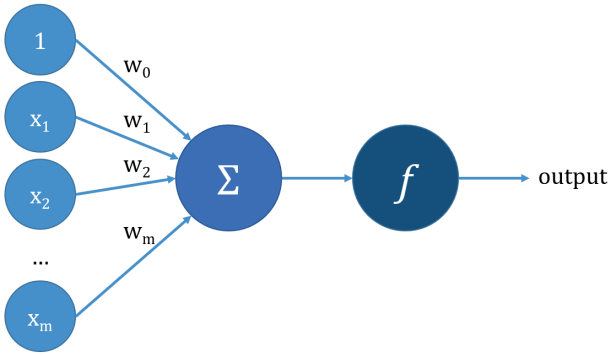

Figure 1.4: Single-layer perceptron, an example of a feedforward neural network that computes a weighted sum of the input followed by a non-linear activation function.

Given a set of N data points $X = \{x^{(i)}\}$ with $x^{(i)} \in \mathbb{R}^C$, and their associated set of target labels $Y = \{y^{(i)}\}$ with $y^{(i)} \in \mathbb{R}^M$, we can train a deep neural network to approximate a mapping function between them. To start with, it is necessary to define a loss or objective function that measures the distance between the network's pre-

dictions $\hat{Y}$ and the labels Y, serving as a proxy for the overall network performance. For regression problems, i.e. tasks were the label is a continuous score, a common choice for the loss function is the p-norm distance. This is a family of continuous and differentiable functions that include, for example, the Manhattan distance ($p = 1$), and the mean squared error function ($p = 2$):

$$L_{p\text{-norm}}(Y, \hat{Y}) = \frac{1}{N} \sum_{i=1}^{N} \left[\left(\sum_{j=1}^{M} \left| y_j^{(i)} - \hat{y}_j^{(i)} \right|^p \right)^{1/p} \right].$$

For applications were the label Y represents a categorical variable, i.e. classes, groups or symbols, we require a well-known function in the information theory literature: the cross-entropy function[30]. Each sample $x^{(i)}$ is associated with a binary code $y^{(i)}$ that indexes one of the possible classes, and the goal of the neural network is to predict the true binary code for the given sample. In this context, the neural network serves as an encoder whose goal is to retrieve the right binary code for a given sample. The cross-entropy loss will be minimum when its value matches the entropy of the true data distribution. In other words, by minimizing the cross-entropy loss we are effectively training an encoder that can reconstruct the true binary codes perfectly:

$$L_{\text{CE}}(Y, \hat{Y}) = \frac{1}{N} \sum_{i=1}^{N} \left[- \sum_{j=1}^{M} y_j^{(i)} \log \hat{y}_j^{(i)} \right].$$

We would like to find the parameters of a network that minimize one of the aforementioned loss functions for a given collection of labeled data points. Although all operations within the neural network are continuous and differentiable, they may be highly non-linear and thus finding an optimal solution analytically is not possible. Furthermore, since these datasets are typically comprised of hundreds of thousands of points, if not millions, with a large input dimensionality, e.g. images, it is unfeasible to load them entirely into memory. With these constraints, stochastic gradient descent (SGD) emerges as a strong candidate method to optimize the values of the network parameters[31].

SGD is an optimization method that minimizes the value of the loss function by iteratively updating the network parameters using randomly sampled subsets of the

```
def train_network(data, network, n_epochs, batch_size,
        cost_function, optimizer, learning_rate):

    # Initialize network parameters
    network.initialize()
    # Initialize optimizer
    optimizer.initialize(learning_rate)

    # Train for a few epochs
    for i in n_epochs:
        # Sample random subset of data
        x, y = data.sample_mini_batch(batch_size)

        # Forward pass: compute predictions and loss
        preds = network.forward(x)
        loss = cost_function(preds, y)

        # Backward pass: compute gradients
        gradients = network.backward(dz=loss.backward(dz=1))

        # Integrate gradients and update network params
        params_update = optimizer.process_grads(gradients)
        network.update(params_update)

    return network
```

Listing 1.1: Pseudo-code for training a neural network.

data. In each step, it follows the opposite direction of the approximate gradients of the loss function with respect to the network parameters. In other words, it calculates the directions of change for the network parameters that minimize the loss function, taking small steps towards them. These gradient values are computed using an algorithm called *backpropagation*. This method calculates the gradient with respect to the parameters as a product of local gradients within each layer:

$$\frac{\partial L}{\partial \theta_i} = \frac{\partial O_i}{\partial \theta_i}\frac{\partial O_{i+1}}{\partial O_i}\cdots\frac{\partial O_K}{\partial O_{K-1}}\frac{\partial L}{\partial O_K},$$

where $\frac{\partial L}{\partial \theta_i}$ represents the gradients of the loss function L with respect to the parameters of layer i; $\frac{\partial O_{i+1}}{\partial O_i}$ is the local partial derivative of the output of layer i+1 with respect to its input; and K indexes the last layer before the loss L. The key of *backpropagation* is that these local derivatives can be computed locally for every layer, and passed upstream as a product of all of the previous derivatives.

Each training step in SGD with *backpropagation* can be decomposed in two operations: the *forward* and the *backward* pass. During the *forward* pass, a subset of the training data is randomly sampled (what is known as the *minibatch*), and fed through the neural network to produce a set of predictions. Note that all intermediate outputs within the network layers are also stored since they are required for gradient calculation. These predictions are compared with the groundtruth labels using the chosen loss function. During the *backward* pass, local gradients are computed starting from the loss function, and progressing upstream until reaching all layers. Finally, all gradients with respect to network parameters are collected and used to update the value of the network parameters.

Despite the remarkable success that gradient-based parameter optimization for deep learning has collected in recent years, there are still drawbacks and opportunities for improvement. Since the gradient values are computed as a product of local derivatives, these values are prone to *vanishing* or *exploding* unless careful measures are taken[32]. A key ingredient to minimize the problem of *vanishing* gradients is the selection of non-linearity function. These functions are applied to the output of every network's intermediate layer to introduce the possibility of non-linear function approximation to the network. While the Sigmoid function was a popular choice a few years ago, its use has been discontinued due to the almost-zero gradient response outside its linear range. Instead, the rectifier linear unit (ReLU) $g(x) = \max(0, x)$ is currently the default choice among deep learning non-linearity functions due to its gradient properties and efficient computation[33]. Other non-linearity functions have been proposed over the years trying to improve upon their predecessors, becoming a very active area of research[34,35] (see Fig. 1.5).

A particularly interesting proposal to reduce the problem of *vanishing* gradients is the idea of residual connections (sometimes also called skip connections). The authors of residual connections propose an architectural change to how layer outputs are computed[36]. Instead of naively applying a layer function f to a given input like $y = f(x)$, they propose to treat the layer function as a residual function added to the input as $y = x + f(x)$ instead. Note how the gradients of the latter term are much more favorable for signal propagation:

$$\begin{aligned} y &= x + f(x) \\ \frac{\partial y}{\partial x} &= 1 + \frac{\partial f(x)}{\partial x}. \end{aligned}$$

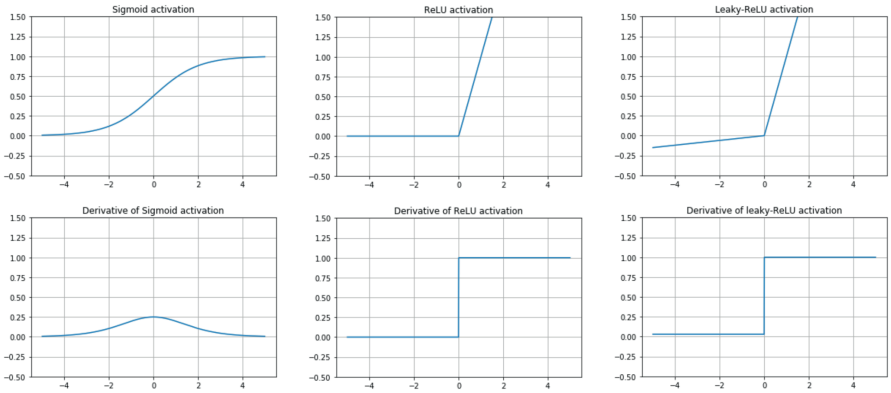

Figure 1.5: Overview of common activation functions and their corresponding derivatives.

This invention allowed to train neural networks composed of thousands of layers successfully. Other improvements have been proposed to accelerate the learning procedure by focusing on the rule used to update the network parameters. Since the gradient calculation performed during SGD with backpropagation is only an approximate measure of the true gradient (computed on a small batch of data points), several authors have proposed techniques to integrate these gradients across training steps. The momentum rule suggests integrating the value of the gradients to enable the optimizer to accelerate when multiple training steps point towards the same direction[37]. Additionally, other authors have proposed to scale the learning rate inversely proportional to the squared norm of the gradients[38], and even combining both momentum and learning rate scaling[39]:

$$\begin{aligned}
\text{Vanilla: } & \theta_{n+1} \leftarrow \theta_n + \alpha\, \partial\theta \\
\text{Momentum: } & m_{n+1} \leftarrow m_n + \beta_1\ (\partial\theta - m_n) \\
\text{RMSProp: } & s_{n+1} \leftarrow s_n + \beta_2\ \left(|\partial\theta|^2 - s_n\right) \\
\text{Adam: } & \theta_{n+1} \leftarrow \theta_n + \frac{\alpha}{s_{n+1}}\, m_{n+1}
\end{aligned}$$

where θ and $\partial\theta$ represent the network parameters and respective gradients; α is the learning rate; and β_1 and β_2 are hyperparameters that control the strength of the momentum and learning rate scaling, respectively.

Finally, batch normalization (BN) has also been praised as a technique that can accelerate training substantially[40]. It keeps a running average of the mean and standard

deviation of the layer input across batch samples, and uses them to normalize the output of the layer to be unit Gaussian. Additionally, it performs a linear mapping of the output with a learned weight and bias. During inference, it uses the statistics computed during training to perform the normalization. It reduces covariate shift and accelerates training by allowing the network to control layer statistics with just two parameters. In practice, it allows to use larger learning rates, substantially reducing training time.

$$
\begin{aligned}
&(1)\ \mu_n \leftarrow \mu_{n-1} + \beta_1 \left(\mu_n - \mu_{n-1}\right) \\
&(2)\ \sigma_n \leftarrow \sigma_{n-1} + \beta_2 \left(\sigma_n - \sigma_{n-1}\right) \\
&(3)\ y = \frac{x - \mu_n}{\sigma_n} \\
&(4)\ z = a\, y + b
\end{aligned}
$$

where x is the input to the layer, y is the normalized input signal, z is the output of the layer, β_1 and β_2 are hyperparameters that control the running averages, and a and b are trainable parameters.

1.5 Convolutional neural networks

Convolutional neural networks (CNNs) are a particularly successful type of deep neural network commonly used in perception tasks such as Computer Vision. Their main characteristic and fundamental difference with fully-connected MLPs is the use of convolutional operations between the input signal and the network parameters[41]. Mathematically, a convolution is a linear operation that performs a sliding dot-product between two signals. Intuitively, it serves as a pattern matching procedure between them, where the output of the convolution operation reaches its maximum when the same pattern appears in both signals.

$$
(F * I)(i, j) = \sum_m \sum_n I(i - m, j - n)\, F(m, n),
$$

where F and I stand for the filter and input signals, respectively, and m and n are the spatial dimensions of I.

A convolutional layer is composed of a set of learnable filters (weights) that perform independent convolutional operations on the input data, resulting in a set of feature maps that are concatenated along the feature dimension and output to the next layer,

see Fig. 1.6. Convolutional layers offer two key advantages over conventional fully connected layers. First, the size of the input data can grow independently from the number of parameters in the network; allowing it to scale the input spatial resolution without exploding the number of trainable parameters. Second, the convolution operation is equivariant to translation, i.e., a translated input produces an equivalent translation in the output signal, which is particularly useful for applications based on object detection.

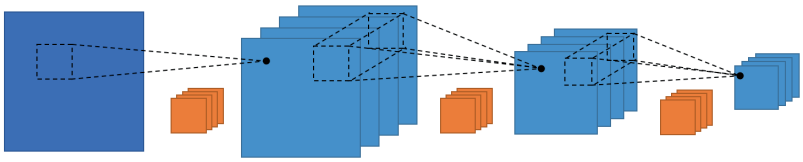

Figure 1.6: Set of convolutional layers with network parameters (filters) in orange, and feature maps in blue.

Convolutional layers can be stacked sequentially (each one followed by an activation function) to create convolutional neural networks. It has been shown that filters in trained networks appear to organize in a hierarchical structure. Early layers, those closer to the network's input, specialize in low-level image features like line or color detection, whereas intermediate or late layers tend to focus on high-level features like part and object detection (see Fig. 1.7).

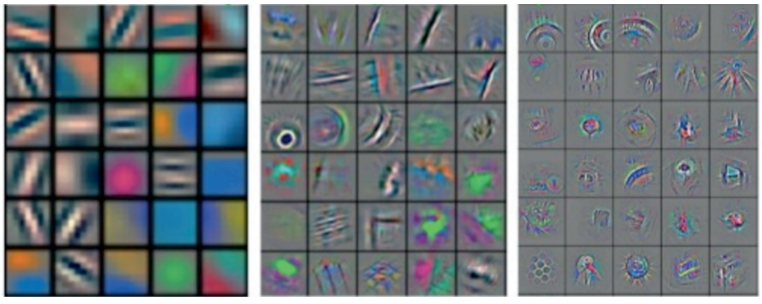

Figure 1.7: Reconstructed patterns from the image samples that cause high activations in a given feature map. While early layers (left) focus on simple patterns, deeper layers (center and right) respond to more elaborated and invariant features. Image credit from Zeiler et al[42].

There exist multiple variations of the vanilla convolution operation depending on the application requirements[43]. For example, strided convolutions reduce the out-

put size along spatial dimensions by skipping some input values during the convolution operation. Taking this idea to the extreme results in the popular max-pooling operation where the filter is a fixed function that takes the maximum value of each local patch. Fractionally-strided convolutions, on the other hand, increase the output size by padding the input signal before the convolution operation, proving useful in generative modeling[44]. Not to confuse with dilated convolutions[45], which pad the filter instead in order to enlarge the receptive field of the CNN, that is, the portion of the input data that has an effect on a certain layer depth and spatial position. Finally, depth-wise separable convolutions decompose the convolution operation into two parts[46]. First, a set of filters is applied to a set of input features maps, pairing each filter with a single feature map. Second, a set of 1×1 filters combine these resulting features maps along the feature dimension, producing the desired output volume. This type of convolution is popular in embedded systems where resources are limited, or low-latency inference is required[47].

1.6 Regularization

The promise of supervised machine learning is that models can automatically find the hidden hypotheses that best explain the relationship between the data and the labels. In real-world applications, these models are trained with a limited amount of data that represents the true data distribution, and we hope that these learned hypotheses can generalize to future unseen data during inference. CNNs are powerful universal function approximators that allow us to discover some of these hypotheses very efficiently, however, this same feature poses a tremendous risk: overfitting to the training data distribution. Overfitting is an undesired phenomenon that consists in learning spurious hypotheses that occur only in the training distribution and that cause the model to underperform in unseen data from the real data distribution. The opposite of overfitting is underfitting, which takes place when the model does not have enough expressive capacity to learn the hidden hypotheses explaining the data-label interaction. With the current high capacity CNNs, underfitting is rarely a problem, whereas addressing overfitting requires significant and specific efforts.

Regularization is a broad concept that consists in performing any kind of modification in the machine learning pipeline that reduces the problem of overfitting, e.g. altering the architecture of the network, the loss function, the input data, or even the training schedule[18]. For example, CNNs are regularized densely-connected neural networks; instead of fully connecting all inputs with trainable neurons, CNNs

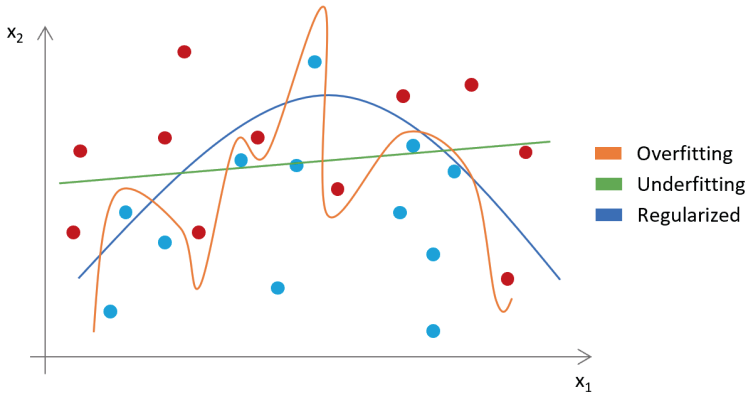

Figure 1.8: Different decision boundaries over a set of data points with two features (horizontal and vertical axes) and a binary label (color). Overfitting, underfitting, and regularized classification models are represented by the orange, green, and blue lines that partition the space.

discover and share feature patterns, resulting in more generalizable filters that reduce overfitting. Other widespread regularization techniques such as weight decay focus on modifying the loss function by adding a regularization term that prevents the model from reaching a global minimum (often associated with overfitting the training set). Weight decay penalizes the loss by adding the norm of the weights, resulting in more evenly distributed weight values that equalize the value of features across the network. Another simple yet effective regularization technique is early stopping, which consists in halting the training procedure when certain condition is met, normally when the performance on the validation data (unseen partition during training) plateaus.

More elaborated forms of regularization modify the network architecture or the input data. Dropout prevents neuron co-adaptation, that is, excessive feature interdependence between neurons, by randomly disabling some of these connections during training[48]. It forces the network to look for alternative hypotheses when some features are unavailable, incentivising redundant connections. Another key regularization technique is data augmentation. Machine learning practitioners often have some prior knowledge about the training data, in particular, they may know certain data transformations that the target labels should be invariant to. In this case, training data can be augmented using these transformations so that the model becomes invariant to them as well. This procedure can be orders of magnitude more resource efficient than manually labeling extra data points, particularly in the med-

ical imaging domain where labels are very expensive[28]. For example, the binary classification problem of detecting human faces in images is invariant to horizontal mirroring. Thus, randomly applying this transformation to training images may be an effective implementation of data augmentation. Augmentation techniques specifically designed for histopathological images are studied in Chapter 3.

1.7 Thesis goals

The main focus of this work is to investigate novel deep learning based methodologies to improve breast cancer prognostic tools within the context of Computational Pathology. This research can be divided into three key blocks:

1. Fundamental challenges in Computational Pathology. We address some of the issues that arise when developing deep learning based models across applications and organs. First, scaling the generation of pixel-level annotated data (Chapter 2). Second, addressing intra- and inter-center stain variation (Chapters 2 and 3). Third, developing accurate and fast models to process entire whole-slide images (Chapters 2 and 4).

2. Automating the core component of breast cancer grading: performing mitosis detection at scale, that is, processing thousands of unseen multicenter entire whole-slide images, while deriving actionable insights for pathologists (Chapter 2).

3. Performing whole-slide image classification. We propose a method that enables feeding entire whole-slide images to a single deep learning based model, targeting patient-level labels and outcome data such as overall survival (Chapters 4 and 5).

1.8 Outline

This thesis is organized as follows:

Chapter 2. We present an algorithm for whole-slide image mitosis detection in H&E breast histology that is robust to stain variation. Furthermore, we develop a novel pipeline to efficiently scale the process of mitotic figure annotation based on automatic analysis of immunohistochemically re-stained slides. Lastly, knowledge distil-

lation was used to reduce the computational requirements of the algorithm.

Chapter 3. We perform a thorough comparison of multiple data augmentation and stain color normalization techniques, and quantify their effects in performance for four common applications of Computational Pathology.

Chapter 4. We propose neural image compression to solve the problem of gigapixel whole-slide image classification. Using an unsupervisely trained neural network encoder, we can drastically reduce the size of whole-slide images while maintaining relevant image-level features. This method allows us to train models that target patient-level labels with only a few hundred data points.

Chapter 5. We extend the idea of neural image compression by training the encoder using a supervised multitask learning approach. By improving the features extracted with the encoder, the method can predict patient-level labels with state-of-the-art performance. Furthermore, the model can learn directly from patient outcome data such as overall survival.

Chapter 6. This chapter discusses the main findings and contributions of this thesis, reflecting on the advancements performed in the field as well as the current limitations. Furthermore, it provides a future outlook for research opportunities in Computational Pathology.

Robust mitosis detection using convolutional neural networks

2

Authors: David Tellez, Maschenka Balkenhol, Irene Otte-Holler, Rob van de Loo, Rob Vogels, Peter Bult, Carla Wauters, Willem Vreuls, Suzanne Mol, Nico Karssemeijer, Geert Litjens, Jeroen van der Laak, and Francesco Ciompi

Original title: Whole-Slide Mitosis Detection in H&E Breast Histology Using PHH3 as a Reference to Train Distilled Stain-Invariant Convolutional Networks

Published in: IEEE Transactions on Medical Imaging (Volume: 37, Issue: 9, Sept. 2018)

DOI URL: doi.org/10.1109/TMI.2018.2820199

Abstract

Manual counting of mitotic tumor cells in tissue sections constitutes one of the strongest prognostic markers for breast cancer. This procedure, however, is time-consuming and error-prone. We developed a method to automatically detect mitotic figures in breast cancer tissue sections based on convolutional neural networks (CNNs).

Application of CNNs to hematoxylin and eosin (H&E) stained histological tissue sections is hampered by noisy and expensive reference standards established by pathologists, lack of generalization due to staining variation across laboratories, and high computational requirements needed to process gigapixel whole-slide images (WSIs). In this chapter, we present a method to train and evaluate CNNs to specifically solve these issues in the context of mitosis detection in breast cancer WSIs.

First, by combining image analysis of mitotic activity in phosphohistone-H3 restained slides and registration, we built a reference standard for mitosis detection in entire H&E WSIs requiring minimal manual annotation effort. Second, we designed a data augmentation strategy that creates diverse and realistic H&E stain variations by modifying H&E color channels directly. Using it during training combined with network ensembling resulted in a stain invariant mitosis detector. Third, we applied knowledge distillation to reduce the computational requirements of the mitosis detection ensemble with a negligible loss of performance.

The system was trained in a single-center cohort and evaluated in an independent multicenter cohort from The Cancer Genome Atlas on the three tasks of the Tumor Proliferation Assessment Challenge. We obtained a performance within the top three best methods for most of the tasks of the challenge.

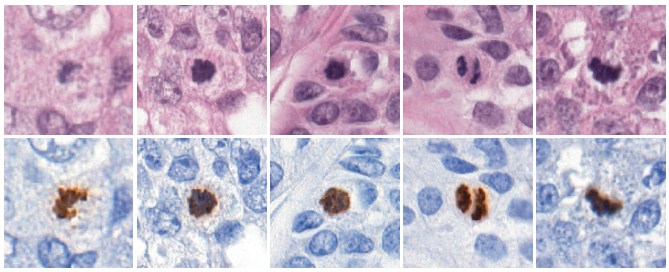

Figure 2.1: Examples of image patches containing mitotic figures, shown at the center of each patch. In H&E (top), mitotic figures are visible as dark spots. In PHH3 (bottom), they are visible as brown spots. Mitotic figures in PHH3 stain are easier to identify than in H&E stain.

2.1 Introduction

Histopathological tumor grade is a strong prognostic marker for the survival of breast cancer patients[11,49]. It is assessed by examination of hematoxylin and eosin (H&E) stained tissue sections using bright-field microscopy[50]. Histopathological grading of breast cancer combines information from three morphological features: (1) nuclear pleomorphism, (2) tubule formation and (3) mitotic count, and can take a value within the 1-3 range, where 3 corresponds to the worst patient prognosis. In this study, we focus our attention on the mitosis count component, as it can be used as a reliable and independent prognostic marker[11,12]. Mitosis is a crucial phase in the cell cycle where a replicated set of chromosomes is split into two individual cell nuclei. These chromosomes can be recognized in H&E stained sections as mitotic figures (see Fig. 2.1). For breast cancer grading, the counting of mitotic figures is performed by first identifying a region of $2\,\mathrm{mm}^2$ with a high number of mitotic figures at low microscope magnification (hotspot) and subsequently counting all mitotic figures in this region at high magnification.

The recent introduction of whole-slide scanners in anatomic pathology enables pathologists to make their diagnoses on digitized slides[51], so-called whole-slide images (WSIs), and promotes the development of novel image analysis tools for automated and reproducible mitosis counting. Publicly available training datasets for mitosis detection[52–55] have important limitations in terms of (1) size, (2) tissue representativity, and (3) reference standard agreement. In these datasets, the total number of annotated mitotic figures is currently limited to 1500 objects, far away from standard datasets used to train modern computer vision systems[56,57]. In addition, mitotic figures were annotated in certain manually selected tumor regions only, often exclud-

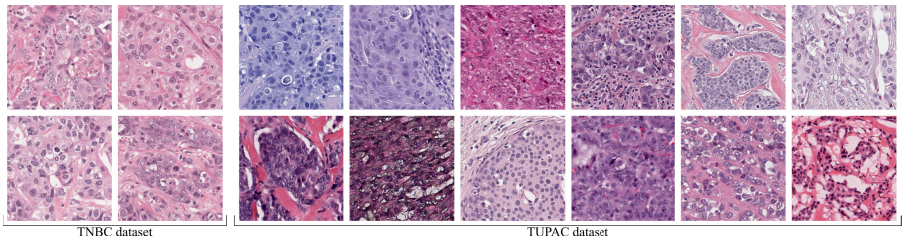

Figure 2.2: Breast tissue samples stained with hematoxylin and eosin (H&E). Each tile comes from a different patient. The triple negative breast cancer (TNBC) cohort contains images from a single center, whereas the Tumor Proliferation Assessment Challenge (TUPAC) dataset contains images from multiple centers. Notice the homogeneous appearance of the *TNBC* dataset, used for training our mitosis detector, and the variable stain of the *TUPAC* dataset, used to validate our method.

ing tissue areas with image artifacts (common in WSIs). Furthermore, exhaustive manual annotations are known to suffer from disagreement among observers and limited recall[58,59]. We propose a method to improve the annotation process based on the automatic analysis of immunohistochemical stained slides. Phosphohistone-H3 (PHH3) is an antibody that identifies cells undergoing mitosis[60,61]. Mitotic figures appear in PHH3 immunohistochemically stained slides (abbreviated as PHH3 stained slides) as high contrast objects that are easier to detect than in H&E[62–64], illustrated in Fig. 2.1. We propose to *destain* H&E slides and *restain* them with PHH3 to obtain both H&E and PHH3 WSIs from the exact same tissue section[65]. By automatically analyzing mitotic activity in PHH3 and registering it to H&E, we generated training data for mitosis detection in H&E WSIs in a scalable manner, i.e. independent from the manual annotation procedure.

Although the process of H&E tissue staining follows a standard protocol, the appearance of the stained slides is not identical among pathology laboratories and varies across time even within the same center (see Fig. 2.2). This variance typically causes mitosis detection algorithms to underperform on images originating from pathology laboratories different than the one that provided the training data[59]. Several solutions have been proposed to tackle this lack of generalization. First, building multicenter training datasets that contain sufficient stain variability. Following this approach, the Tumor Proliferation Assessment Challenge (TUPAC) resulted in numerous successful mitosis detection algorithms. Top-performing methods in the challenge are based on convolutional neural networks (CNNs)[18,58,59,66,67]. This is in line with the trend observed in recent years, which has seen CNNs as the top-

performing approach in image analysis, both in computer vision and medical imaging[28], and corroborates the fact that CNNs have become the standard methodology for automatic mitosis detection. However, multicenter datasets cannot cover all the variability encountered in clinical practice, and are expensive to collect. Second, stain standardization techniques[68–70] have been widely used by many of these successful mitosis detection methods to reduce stain variability. However, they require preprocessing all training and testing WSIs and do not reduce the generalization error of trained models. Third, data augmentation strategies have been used to simulate stain variability during the model training. These techniques typically involve RGB transformations such as brightness and contrast enhancements, and color hue perturbations[59,71]. We argue that designing specific data augmentation strategies for H&E stained tissue images is the most promising approach to reduce the generalization error of these networks, avoiding the elevated costs of assembling a multicenter cohort, and effectively enforcing stain invariance into the trained models. We propose an augmentation strategy tailored to H&E WSIs that modifies the hematoxylin and eosin color channels directly, as opposed to RGB, and it is able to generate a broad range of realistic H&E stain variations from images originating in a single center. We call this technique *stain augmentation*.

Automatic mitosis detection algorithms rely on techniques such as the use of high capacity CNNs and multi-network ensembling to achieve state of the art performance[55,58,66]. These are simple yet effective mechanisms to improve performance, reduce generalization error and diminish the sensitivity of the model to the detection threshold. However, due to their computationally expensive nature, it is unfeasible to use them for dense prediction in gigapixel WSIs. We propose to exploit the technique of *knowledge distillation*[72] to reduce the size of the trained ensemble to that of a single network, maintaining similar levels of performance and increasing processing speed drastically.

Our contributions can be summarized as follows:

- We propose a scalable procedure to exhaustively annotate mitotic figures in H&E WSIs with minimal human labeling effort. We do so by automatically analyzing mitotic activity in PHH3 restained tissue sections and registering it to H&E.

- We propose a data augmentation technique that generates a broad range of realistic H&E stain variations by modifying the hematoxylin and eosin color channels directly. We demonstrate its ability to enforce stain invariance by

transferring the performance obtained in a dataset from a single center to a multicenter publicly available cohort.

- We apply knowledge distillation to reduce the size of an ensemble of trained networks to that of a single network, in order to perform mitosis detection in gigapixel WSIs with similar performance and vastly increased processing speed.

The chapter is organized as follows. Sec. 2.2 reports the datasets used to train and validate our method. Sec. 2.3 and Sec. 2.4 describe the methodology in depth. All details regarding CNN architectures, training protocols and hyper-parameter tuning are explained in Sec. 2.5. Experimental results are listed in Sec. 2.6, followed by Sec. 2.7 where the discussion and final conclusion are stated.

2.2 Materials

In this study, we use two cohorts from different sources for (1) developing the main mitosis detection algorithm, and (2) performing an independent evaluation of the system performance. Details on the datasets are provided in Table 2.1.

The first cohort consists of 18 triple negative breast cancer (TNBC) patients who underwent surgery in three different hospitals in the Netherlands: Jeroen Bosch Hospital, Radboud University Medical Centre (Radboudumc) and Canisius-Wilhelmina Hospital. However, all tissue sections were cut, stained and scanned at the Radboudumc using a 3DHistech Pannoramic 250 Flash II scanner at a spatial resolution of $0.25\,\mu\mathrm{m/pixel}$, therefore, we consider this set of WSIs a single-center one. Subsequently, the slides were destained, restained with PHH3 and re-scanned, resulting in 18 pairs of H&E and PHH3 WSIs representing the exact same tissue section per pair. We will refer to these images as the *TNBC-H&E* and *TNBC-PHH3* datasets through the rest of the chapter.

The second cohort consists of the publicly available TUPAC dataset[55]. In particular, 814 H&E WSIs from invasive breast cancer patients from multiple centers included in The Cancer Genome Atlas[73] scanned at $0.25\,\mu\mathrm{m/pixel}$ were annotated, providing two labels for each case. The first label is the histological grading of each tumor based on the mitotic count only. The second score is the outcome of a molecular test highly correlated with tumor proliferation[74]. Out of these 814 WSIs, 493 cases have a public reference standard, whereas the remaining 321 cases do not (only available

Table 2.1: Overview of the datasets used in this study. The purpose of *training* indicates that the dataset was used to train a CNN for mitosis detection, whereas *threshold* tuning indicates that it was solely employed to optimize certain hyper-parameters, such as the detection threshold.

Alias	Data	Cases	Multicenter	Reference standard	Purpose
TNBC-H&E	H&E whole-slide images	18	No	None	Training
TNBC-PHH3	PHH3 whole-slide images	18	No	Set of annotated patches	Training
TUPAC-train	H&E whole-slide images	493	Yes	Grading and proliferation score	Threshold tuning
TUPAC-aux-train	H&E selected regions	50	Yes	Exhaustive location of mitotic figures	Threshold tuning
TUPAC-test	H&E whole-slide images	321	Yes	Not publicly available	Evaluation
TUPAC-aux-test	H&E selected regions	34	Yes	Not publicly available	Evaluation

to TUPAC organizers). We will refer to these sets of WSIs as the *TUPAC-train* and *TUPAC-test* datasets respectively.

Additionally, the organizers of TUPAC provided data for individual mitotic figure detection from two centers. They preselected 84 proliferative tumor regions from H&E WSIs of breast cancer cases, and two observers exhaustively annotated them. Coincident annotations were marked as the reference standard, and discording items were reviewed by a third pathologist. Out of these 84 regions, 50 cases have a public reference standard, whereas the remaining 34 cases do not (only available to TUPAC organizers). We will refer to these sets as the *TUPAC-aux-train* and *TUPAC-aux-test* datasets respectively.

All WSIs in this work were preprocessed with a tissue-background segmentation algorithm [75] in order to exclude areas not containing tissue from the analysis.

2.3 PHH3 stain: reference standard for mitotic activity

We trained a CNN to automatically identify mitotic figures throughout PHH3 WSIs and registered the detections to the H&E WSI pairs. The entire process is summarized in Fig. 2.3.

2.3.1 Mitosis detection in PHH3 whole-slide images

We used color deconvolution to disentangle the DAB and hematoxylin stains (brown and blue color channels respectively) and obtained the mitotic candidates by labeling connected components of all positive pixels belonging to the DAB stain. We observed that the PHH3 antibody was sensitive but not specific regarding mitosis activity: 75% of candidates were trivial artifacts. To avoid waste of manual annotation effort on these artifacts, we trained a CNN, named *CNN1*, to classify candidates among artifactual or non-artifactual objects. To train such a system, a student labeled 2000 randomly selected candidates. We classified all PHH3 candidates with *CNN1*, and randomly selected 2000 samples classified as non-artifactual. Four observers labeled these samples as either containing a mitotic figure or not. These four observers consisted of a pathology resident, a PhD student and two lab technicians, and they were provided with sufficient training, visual examples and hands-on practice on mitosis detection. Annotations were aggregated by performing majority voting,

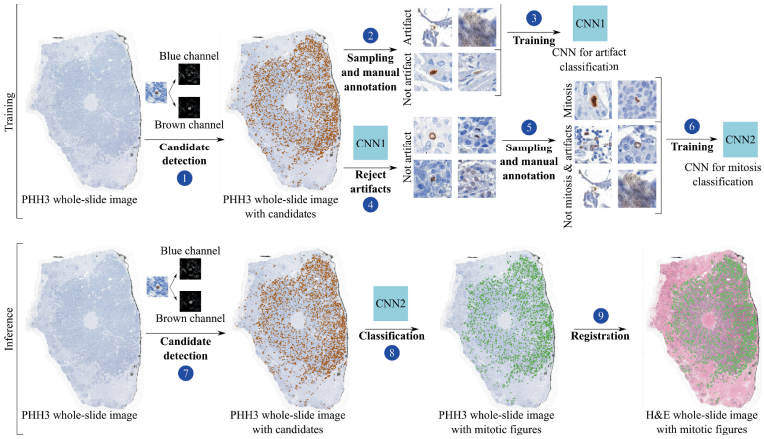

Figure 2.3: Building reference standard for mitotic activity using PHH3 stained slides. Top: training stage, where mitotic candidates are extracted from the brown color channel (1), pruned from artifacts (2, 3, 4), a subset is manually annotated (5), and then used to train a CNN to distinguish between mitotic and non-mitotic patches (6), named *CNN2*. Bottom: inference stage, where candidates in a given PHH3 slide (7) are classified with *CNN2* as mitotic or non-mitotic (8), then registered to their respective H&E slide pairs (9).

keeping only those samples where at least 3 observers agreed upon. This resulted in 778 and 1093 mitotic and non-mitotic annotations, respectively. Furthermore, the non-mitotic set was extended with 1500 artifactual samples used to train *CNN1*, resulting in 2593 non-mitotic samples.

We used this annotated set of samples to train *CNN2* to distinguish PHH3 candidates among mitotic and non-mitotic patches. During training, we randomly applied several techniques to augment the data and prevent overfitting, namely: rotations, vertical and horizontal mirroring, elastic deformation[76], Gaussian blurring, and translations. The resulting performance of *CNN2* was an F1-score of 0.9. Details on network architecture, training protocol and hyper-parameter selection are provided in Sec. 2.5. We classified all candidates found in the PHH3 slides as mitotic or non-mitotic objects using *CNN2*, generating exhaustive reference standard data for mitosis activity at whole-slide level.

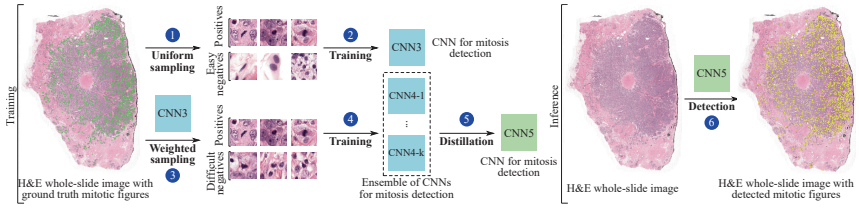

Figure 2.4: Training a whole-slide H&E mitosis detector. Left (training stage): an auxiliary network *CNN3* is trained with uniformly sampled patches (1, 2). Then, *CNN3* is used to perform hard mining on the negative patches and create k distinct datasets sampled via bootstrapping (3). These datasets are used to train networks *CNN4-1* to *CNN4-k* and build an ensemble (4). Finally, the ensemble is reduced into *CNN5* via knowledge distillation (5). Right (inference stage): *CNN5* is applied in a fully convolutional manner throughout an entire WSI to detect mitotic figures exhaustively (6).

2.3.2 Registering mitotic figures from PHH3 to H&E slides

The process of slide restaining guaranteed that the exact same tissue section was present in both the H&E and the PHH3 WSIs, requiring minimal registration to align mitotic objects. We designed a simple yet effective two-step routine to reduce vertical and horizontal shift at *global* and *local* scale. First, we performed a global and coarse registration that minimized the vertical and horizontal shift between image pairs. We did so by finding the alignment of randomly selected pairs of corresponding tiles as the shift vector that maximized the 2D cross-correlation. For improved accuracy, we repeated this procedure 10 times per WSI pair (at random locations throughout the WSI), averaging the cross-correlation heatmap across trials. Finally, all mitoses were adjusted with the resulting global shift vector. Second, we registered each mitotic figure individually, following a similar procedure as before. We extracted a single pair of high magnification tiles, centered in each candidate location, to account for individual local shifts.

2.4 H&E stain: training a mitosis detector

We trained a CNN for the task of mitosis detection and used it to exhaustively locate mitotic figures throughout H&E WSIs. Only slides from the *TNBC-H&E* dataset were used in this step. The procedure is summarized in Fig. 2.4.

2.4.1 Assembling a training dataset for mitosis detection

Even though the *TNBC-H&E* dataset already possessed little stain variation, we standardized the stain of each WSI to reduce intra-laboratory stain variations[70], preventing the CNN from becoming stain invariant from the raw training data. This further strengthens the challenge of generalizing to unseen stain variations.

As a result of the large amount of available pixels in WSIs, the selection of negative samples was not trivial. We propose a *candidate detector* based on the assumption that mitotic figures are non-overlapping circular objects with dark inner cores. This detector found candidates by iterative detection and suppression of all circular areas of diameter d centered on local minima of the mean RGB intensity until all pixels above a threshold t were exhausted. We selected a sufficiently large t so that candidates were representative for all tissue types. A candidate was labeled as a positive sample if its Euclidean distance to any reference standard object was at most d pixels, and labeled as a negative sample otherwise.

Most of the negative samples were very easy to classify and their contribution to improve the decision boundaries of the CNN was marginal. We found it crucial to identify highly informative negative samples to train the CNN effectively. We proceeded similarly as stated in[58]. First, we built an *easy* training dataset by including all positive candidates and a number of uniformly sampled negative candidates, and trained a network with it, labeled as *CNN3* for future reference. Second, we evaluated all candidate patches with this network, obtaining a prediction probability for each of them. Finally, we built a *difficult* training dataset by selecting all positive candidates, and a number of negative candidates sampled proportionally to their probability of being mitosis, so that harder samples were chosen more often.

2.4.2 H&E stain augmentation

We trained a CNN on the *difficult* dataset to effectively distinguish between mitotic and non-mitotic H&E patches, named as *CNN4*. During training, we applied several techniques to augment the dataset on-the-fly, preventing overfitting and improving generalization. We implemented several routines for H&E histopathology imaging in the context of mitosis detection, illustrated in Figure 2.5 with a sample patch.

Morphology invariance. We exploited the fact that mitotic figures can have variable shapes and sizes by augmenting the training patches with rotation, vertical and hor-

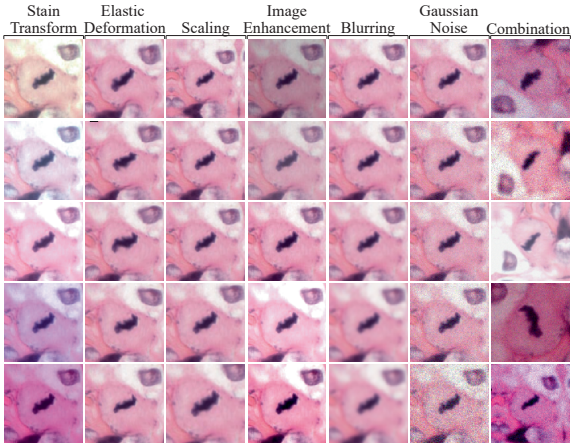

Figure 2.5: Multiple augmented versions of the same mitotic patch. Each column shows samples of a single augmentation function except for the last one, which combines all the techniques together with rotation and mirroring.

izontal mirroring (*R*), scaling (*S*), elastic deformation (*E*)[76], and translation around the central pixel.

Stain invariance. We used a novel approach to simulate a broad range of realistic H&E stained images by retrieving and modifying the intensity of the hematoxylin and eosin color channels directly (*C*), as illustrated in Figure 2.6. First, we transformed each patch sample from RGB to H&E color space using a method based on color deconvolution[77], see the Appendix for more methodological details. Second, we modified each channel i individually, i.e. hematoxylin (H_{ch}), eosin (E_{ch}) and residual (R_{ch}), with random factors α_i and biases β_i taken from two uniform distributions. Finally, we combined and projected the resulting color channels back to RGB. Additionally, we simulated further alternative stains by modifying image brightness, contrast and color intensity (*H*).

Artifact invariance. We mimicked the out of focus effect of whole-slide scanners with a Gaussian filter (*B*), and added Gaussian noise to decrease the signal-to-noise ratio of the images (*G*), simulating image compression artifacts.

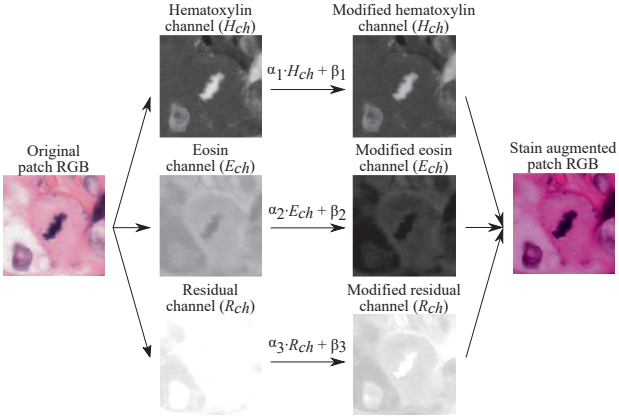

Figure 2.6: H&E stain augmentation. From left to right: first, an RGB patch is decomposed into hematoxylin (H_{ch}), eosin (E_{ch}) and residual (R_{ch}) color channels. Then, each channel is individually modified with a random factor and bias. Finally, resulting channels are transformed back to RGB color space.

2.4.3 Ensemble & network distillation

The use of an ensemble of networks is a key factor to achieve state of the art performance in multiple classification tasks[57], particularly in mitosis detection[58,66]. An ensemble of networks performs significantly better than any of its members if they make independent (uncorrelated) errors. Building different training datasets with replacement (bagging) has been shown to increase model independence in an ensemble[18]. Therefore, we trained k different CNNs on k different training datasets, each one obtained by sampling negative candidates with replacement, and made an ensemble with the networks, averaging their predicted probabilities across models.

The computational requirements of this ensemble grow proportionally to k. To reduce this burden, we applied the idea of knowledge distillation, a technique designed to transfer the performance of a network (or ensemble of networks) to a lighter target neural network[72]. To achieve the highest performance, we distilled the ensemble of k networks to a single smaller CNN, named *CNN5*. We did so by training *CNN5* directly on the continuous averaged output probabilities of the ensemble, instead of the dataset labels, as indicated in[72]. We defined γ as a parameter to control the amount of trainable parameters used by *CNN5*, taking values in the $[0, 1]$ range. In particular, the number of filters per convolutional layer was proportional to this parameter. It defaults to $\gamma = 1$, unless stated otherwise.

2.4.4 Outcome at whole-slide level

We detected mitotic figures at whole-slide level by sliding *CNN5* over tissue regions at $0.25\,\mu\mathrm{m/pixel}$ resolution, producing a mitosis probability map image for each slide. Simple post-processing allowed to detect individual mitotic objects: (1) thresholding and binarizing the probability map (pixel probability of at least 0.8 to reduce the computational burden), (2) labeling connected components in the resulting image, and (3) suppressing double detections closer than d pixels. A detection probability threshold δ must be provided to discern between false and true positives.

In order to provide the number of mitotic figures in the most active tumor area per slide, we slid a virtual hotspot consisting of a $2\,\mathrm{mm}^2$ circle throughout each entire WSI, counting the number of mitoses at each unique spatial position. To identify the hotpot with the largest mitotic activity ignoring large outliers, we considered the 95th percentile of the series of mitotic countings for each slide, excluding empty areas.

To estimate the tumor grading of the patient, we followed the guidelines used to compute the Mitotic Activity Index (MAI)[50]. We defined two thresholds, θ_1 and θ_2, and used them to categorize the number of mitotic figures in the hotspot into three possible outcomes. In particular, we predicted grade 1, 2 or 3 depending on whether the mitotic counting was below or equal to θ_1, between thresholds, or above θ_2, respectively. To estimate a continuous tumor proliferation score, we simply provided the number of mitotic figures in the hotspot.

2.5 CNN architecture, training protocol and other hyperparameters

In order to train the CNN models, we used RGB patches of 128×128 pixel size, taken at $0.25\,\mu\mathrm{m/pixel}$ resolution and whose central pixel was centered in the coordinates of the annotated object. Patches were cropped as part of the data augmentation strategy, resulting in 100×100 pixel images fed into the CNNs.

Convolutional neural networks were trained to minimize the cross-entropy loss of the network outputs with respect to the patch labels, using stochastic gradient descent with Adam optimization and balanced mini-batch of 64 samples. To prevent overfitting, an additional L_2 term was added to the network loss, with a fac-

tor of 1×10^{-5}. Furthermore, the learning rate was exponentially decreased from 1×10^{-3} to 3×10^{-5} through 20 epochs. At the end of training, network parameters corresponding to the checkpoint with the highest validation F1-score were selected for inference.

2.5.1 Mitosis detection in PHH3

CNN1 was trained with 1500 artifactual and 500 non-artifactual samples, and *CNN2* was trained with 778 mitotic and 2593 non-mitotic patch samples. In both cases, the sets of samples were randomly divided into training and validation subsets at case level from *TNBC-PHH3*, with 10 and 8 slides for training and validation, respectively. The architecture of *CNN1* and *CNN2* consisted of five pairs of convolutional and max-pooling layers with 64, 128, 256, 512 and 1024 3×3 filters per layer, followed by two densely connected layers of 2048 and 2 units respectively. A dropout layer was placed between the last two layers, with 0.5 coefficient. All convolutional and dense layers were followed by ReLU functions, except for the last layer that ended with a softmax function.

Table 2.2: Architecture of *CNN3*, *CNN4* and *CNN5*. γ controls the number of filters per convolutional layer.

Function	Filters	Size	Stride	Activation
conv	32 γ	3x3	1	Leaky-ReLU
conv	32 γ	3x3	2	Leaky-ReLU
conv	64 γ	3x3	1	Leaky-ReLU
conv	64 γ	3x3	2	Leaky-ReLU
conv	128 γ	3x3	1	Leaky-ReLU
conv	128 γ	3x3	1	Leaky-ReLU
conv	256 γ	3x3	1	Leaky-ReLU
conv	256 γ	3x3	1	Leaky-ReLU
conv	512 γ	14x14	1	Leaky-ReLU
dropout	-	-	-	-
conv	2	1	1	Softmax

2.5.2 Mitosis detection in H&E

The H&E slides provided in the *TNBC-H&E* dataset were randomly divided into training, validation and test subsets, with 11, 3 and 4 slides each. For the candidate detector, we selected $d = 100$ as the diameter of an average tumor cell at $0.25\,\mu\mathrm{m/pixel}$ resolution; and $t = 0.6$ recalling 99% of the mitotic figures in the reference standard of the validation set, with a rate of 1 positive to every 1000 negative samples. On average, each slide had 1 million candidates. The architecture of *CNN3*, *CNN4* and *CNN5* is summarized in Table 2.2. We found that substituting max-pooling layers for strided convolutions slightly improved convergence, resulting in an *all convolutional* architecture[78]. To train *CNN3*, we built a training set with all positive samples, $22{,}000$ mitotic figures, and $100{,}000$ uniformly sampled negative candidates. For validation purposes, we also built a validation set consisting of 10% of the total available samples in both classes, $200{,}000$ negative and 500 positive candidates. To train *CNN4* and *CNN5*, we built a training set with all positive samples, $22{,}000$ mitotic figures, and $400{,}000$ negative candidates, sampling difficult patches more often with replacement. We used the same validation set as with *CNN3*. For the ensemble, we selected $k = 10$, the highest number of networks that we could manage in an ensemble with our computational resources. We distilled several versions of *CNN5* varying the value of γ, evaluated each network in the validation set of the *TNBC-H&E* dataset, and obtained an F1-score of 0.634, 0.646 and 0.629 for γ values of 1.0, 0.8 and 0.6, respectively. We selected $\gamma = 0.6$ for further experiments, resulting in a distilled network with 28X and 2.8X times less parameters than the ensemble and the single network ($\gamma = 1.0$), respectively, at a negligible cost of performance.

The color augmentation technique sampled α and β from two uniform distributions with ranges [0.95, 1.05] and [-0.05, 0.05], respectively. Patches were scaled with a zooming factor uniformly sampled from [0.75, 1.25]. The elastic deformation routine used $\alpha = 100$, and $\sigma = 10$. Color, contrast and brightness intensity was enhanced by factors uniformly sampled from [0.75, 1.5], [0.75, 1.5] and [0.75, 1.25], respectively. The Gaussian filter used for blurring sampled σ uniformly from the [0, 2] range. The additive Gaussian noise had zero mean and a standard deviation uniformly sampled from the [0, 0.1] range. These parameters were selected empirically to simultaneously maximize visual variety and result in realistic samples.

2.6 Experimental results

2.6.1 Impact of augmentation, ensembling and distillation

We performed a series of experiments to quantitatively assess the impact in performance of three ideas mentioned in this chapter: (a) data augmentation, (b) ensemble and (c) knowledge distillation. In each experiment, we trained a CNN (or set of CNNs for the ensemble case) with the *TNBC-H&E* dataset as explained in Sec. 2.4 and 2.5. Each trained model was evaluated in the independent *TUPAC-aux-train* dataset with multiple detection thresholds, reporting the highest F1-score obtained. Table 2.3 summarizes the numerical results, and Figure 2.7 analyzes the sensitivity of each model with respect to the detection threshold.

Table 2.3: Analysis of the impact in performance of using data augmentation, ensemble and knowledge distillation. Each row represents a CNN (or set of CNNs for the ensemble case) that was trained using *TNBC-H&E* data as explained in Sec. 2.4. Each trained network was evaluated in the independent *TUPAC-aux-train* dataset with multiple detection thresholds, reporting the highest F1-score obtained and the number of trainable parameters. Experiments 1, 2 and 3 compared the use of different data augmentation strategies (R: rotation, C: color stain, S: scaling, etc., see Sec. 2.4 for the full list). Experiment 4 showed the performance of an ensemble of $k = 10$ networks, trained as explained in Sec. 2.4. In experiments 5, 6 and 7, the ensemble of CNNs (experiment 4) was distilled into single smaller CNNs with varying capacities $\gamma = 1.0, 0.8, 0.6$ as explained in Sec. 2.4. The CNN trained for experiment 7 coincides with *CNN5*.

Exp	Augment	Ensemble	Distilled	F1-score	Param
1	RSEB	No	No	0.018	26.9M
2	RCSEB	No	No	0.412	26.9M
3	RCSEHBG	No	No	0.613	26.9M
4	RCSEHBG	$k = 10$	No	0.660	269M
5	RCSEHBG	No	$\gamma = 1.0$	0.623	26.9M
6	RCSEHBG	No	$\gamma = 0.8$	0.628	17.1M
7	RCSEHBG	No	$\gamma = 0.6$	0.636	9.5M

Data augmentation. The goal of experiments 1, 2 and 3 is to test whether the proposed data augmentation strategy can improve the performance of the CNN in an independent test set, in particular the novel color stain augmentation. In experiment 1, we trained a baseline system including only basic augmentation (*RSEB*) and ob-

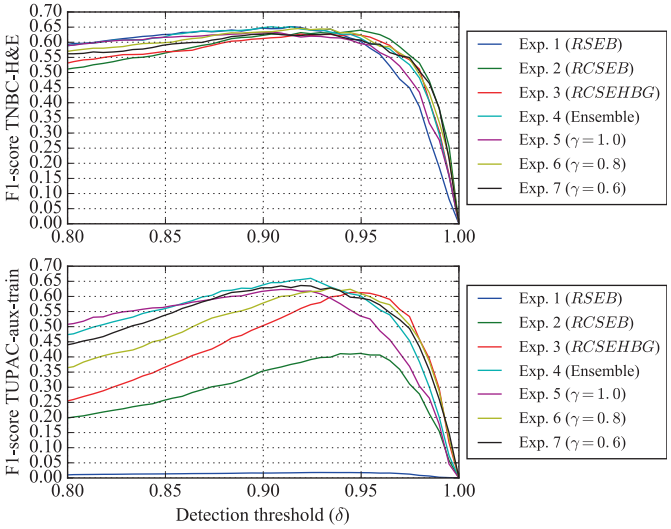

Figure 2.7: Analysis of the impact in performance of using data augmentation, ensemble and knowledge distillation measured in terms of F1-score with respect to the detection threshold (top: *TNBC-H&E dataset*, bottom: *TUPAC-aux-train dataset*). Experiments 1, 2 and 3 compared the use of different data augmentation strategies (R: rotation, C: color stain, S: scaling, etc., see Sec. 2.4 for the full list). Experiment 4 showed the performance of an ensemble of $k = 10$ networks, trained as explained in Sec. 2.4. In experiments 5, 6 and 7, the ensemble of CNNs (experiment 4) was distilled into single smaller CNNs with varying capacities $\gamma = 1.0, 0.8, 0.6$ as explained in Sec. 2.4. The CNN trained for experiment 7 coincides with *CNN5*.

tained an F1-score of 0.018. In experiment 2, we repeated the training procedure including our color augmentation technique as well (*RCSEB*) and obtained an F1-score of 0.412. In experiment 3, we repeated the training procedure including all the augmentation techniques mentioned in Sec. 2.4 (*RCSEHBG*) and obtained an F1-score of 0.613.

Ensemble. The goal of experiment 4 is to test whether the use of an ensemble of networks can improve the performance of the mitosis detector beyond the results obtained in experiment 3 with a single network. We trained and combined a set of CNNs, as explained in Sec. 2.4, and obtained an F1-score of 0.660. Furthermore, we analyzed its performance with respect to the detection threshold and observed a more robust behavior than that of the single CNN tested in experiment 3, illustrated in Fig. 2.7

Distillation. The goal of experiments 5, 6 and 7 is to test whether knowledge distillation can effectively transfer the performance of the ensemble trained in experiment 4 to a single CNN. We trained three CNNs with γ set to 1.0, 0.8 and 0.6, respectively to experiments 5, 6 and 7. They all exhibited a similar performance to that of the ensemble (F1-score of 0.623, 0.628 and 0.636, respectively), with drastically less trainable parameters and superior performance to a single CNN trained without distillation (experiment 3). Experiment 7 resulted in *CNN5*, used in the following sections.

2.6.2 Comparison with the state of the art

We evaluated the performance of our system in the three tasks of the TUPAC Challenge[55], and compared the results with those of top-performing teams, summarized in Tab. 2.4.

Table 2.4: Independent evaluation of the proposed method performance in the three tasks of the TUPAC challenge. Columns Top-1, Top-2 and Top-3 correspond to the best performing solutions in the public leaderboard, respectively.

Dataset	Ground truth	Metric	Top-1	Top-2	Top-3	Proposed [95 c.i.]
TUPAC-test	Tumor grading	Kappa	0.567	0.534	0.462	0.471 [0.340, 0.603]
TUPAC-test	Proliferation score	Spearman	0.617	0.516	0.503	0.519 [0.477, 0.559]
TUPAC-aux-test	Mitosis location	F1-score	0.652	0.648	0.616	0.480

We used *CNN5* with $\gamma = 0.6$ for all submissions, solely trained with the *TNBC-H&E* dataset. Notice that the authors do not have access to the ground truth data of *TUPAC-test* and *TUPAC-aux-test*. Our model predictions were independently evaluated by the organizers of TUPAC. This ensured fair and independent comparison with state of the art methods. For the first and second tasks, we tuned the hyper-parameters of the proposed whole-slide mitosis detector with the *TUPAC-train* dataset. We selected $\delta = 0.970$ to maximize the Spearman correlation between our mitotic count prediction and the ground truth proliferation score. Then, we tuned θ_1 and θ_2 to maximize the quadratic weighted Cohen's kappa coefficient between our predicted tumor grade and the ground truth, obtaining $\theta_1 = 6$ and $\theta_2 = 20$. For the third task, we selected $\delta = 0.919$ to maximize the F1-score metric in the *TUPAC-aux-train* dataset. Spearman, kappa and F1-score are the evaluation metrics proposed in the TUPAC Challenge.

For the first task, we obtained a Kappa agreement of 0.471 with 95 confidence intervals [0.340, 0.603] between the ground truth tumor grading and our prediction on the *TUPAC-test* dataset. This performance is comparable to the top-3 entry in the leaderboard.

For the second task, we obtained a Spearman correlation of 0.519 with 95 confidence intervals [0.477, 0.559] between the ground truth genetic-based proliferation score and our prediction on the *TUPAC-test* dataset. This performance is comparable to the top-2 entry in the leaderboard.

For the third task, we obtained an F1-score of 0.480, with a precision and recall values of 0.467 and 0.494, respectively, by detecting individual mitotic figures in the *TUPAC-aux-test*. This performance is comparable to the top-7 entry in the leaderboard.

2.6.3 Observer study: precision of the mitosis detector

Due to the relatively low F1-score of 0.480 obtained in the *TUPAC-aux-test*, compared to the F1-score of 0.636 obtained in the *TUPAC-aux-train*, we investigated whether this difference could potentially be caused by a combination of inter-observer variability in the TUPAC reference standard, which was established by human observers, and lack of sufficient number of test samples. A resident pathologist manually classified the detections of *CNN5* on the *TUPAC-aux-test*, blinded to the patch labels. The observer indicated that 128 out of 181 detections contained mitotic figures, resulting in a precision of 0.707 for the detector. With this precision and assuming the recall suggested by the organizers, we would obtain an alternative F1-score of 0.581 in the *TUPAC-aux-test*. For the sake of completeness, the patches used in this experiment are depicted in the Appendix (Fig. 2.8).

2.7 Discussion and conclusion

To the best of our knowledge, this is the first time that the problem of noisy reference standards in training algorithms for mitosis detection in H&E WSIs was solved using immunohistochemistry. We validated our hypothesis that mitotic activity in PHH3 can be exploited to train a mitosis detector for H&E WSIs that is competitive with the state of the art. We proposed a method that combined (1) H&E-PHH3 restaining, (2)

automatic image analysis and (3) registration to exhaustively annotate mitotic figures in entire H&E WSIs, for the first time. Only 2 hours of manual annotations per observer were needed to train the algorithm, delivering a dataset that was at least an order of magnitude larger than the publicly available one for mitosis detection. Using this method, the total number of annotated mitotic figures in H&E was solely limited by the number of restained slides available, not the amount of manual annotations. This is a very desirable property in the Computational Pathology field where manual annotations require plenty of resources and human expertise. Our work serves as a proof of concept to show that the combination of restaining, image analysis and registration can be used to automatically generate ground truth at scale when immunohistochemistry is the reference standard.

Staining variation between centers has long prohibited good generalization of algorithms to unseen data. In this work we applied a stain augmentation strategy that modifies the hematoxylin and eosin color channels directly, resulting in training samples with diverse and realistic H&E stain variations. Our experimental results indicate that the use of H&E-specific data augmentation and an ensemble of networks were key ingredients to drastically reduce the CNN's generalization error to unseen stain variations, i.e. transferring the performance obtained in WSIs from a single center to a cohort of WSIs from multiple centers. Furthermore, these results suggest that it is possible to train robust mitosis detectors without the need for assembling multicenter training cohorts or using stain standardization algorithms. More generally, we think that this combination of H&E-specific data augmentation and ensembling could benefit other applications where inference is performed on H&E WSIs, regardless of the tissue type.

High capacity CNNs typically exhibit top performance in a variety of tasks in the field of Computational Pathology, including mitosis detection. However, they come with a computational burden that can potentially compromise their applicability in daily practice. By using knowledge distillation, we massively reduced the computational requirements of the trained detector at inference time. In particular, we shrank the size of the distilled model 28 times (see Tab. 2.3), with a negligible performance loss. This reduction combined with the fully convolutional design of the distilled network enabled us to perform efficient dense prediction at gigapixel-scale, processing entire TUPAC WSIs at $0.25\,\mu\mathrm{m/pixel}$ resolution in 30-45 min.

On the task of individual mitosis detection in the *TUPAC-aux-test* set, we obtained different precision scores from the TUPAC organizers (0.467) and our observer (0.707).

We attribute this disagreement to two factors: (1) the method used to annotate the images, and (2) the small number of samples in the test set. According to TUPAC organizers, two pathologists independently traversed the image tiles and identified mitotic figures. Coincident detections were marked as reference standard, and discording items were reviewed by a third pathologist. Notice that mitotic figures missed by both pathologists were never reported, potentially resulting in true mitotic figures not being annotated. This lack of recall could explain the high number of false positives initially detected by our network and later found to be true mitotic figures by an expert observer. Due to the small number of samples in the *TUPAC-aux-test* set (34 tiles), this effect can cause a large distortion in the F1-score. For reproducibility considerations, we have included all detections in the Appendix (Fig. 2.8). These results illustrate the difficulty of annotating mitotic figures manually, specifically in terms of recalling them throughout large tissue regions, and supports the idea of using PHH3 stained slides as an objective reference standard for the task of mitosis detection.

As a limitation of our method, we acknowledge the existence of some noise in the reference standard generated by analyzing the PHH3 WSIs and attribute it to three components: (1) the limited sensitivity of the PHH3 antibody (some late-stage mitotic figures were not highlighted, thus not even considered as candidates); (2) the limited specificity of the PHH3 antibody (many of the candidates turned out to be artifacts); and (3) the limited performance of *CNN2* (F1-score of 0.9). This noise restricted our ability to detect small performance changes during training, potentially resulting in suboptimal model and/or hyper-parameter choices. More carefully PHH3 restaining process and improved training protocols could palliate this effect in the future.

In terms of future work, mitotic density at whole-slide level could be exploited to find the location of the tumor hotspot, potentially resulting in significant speedups in daily practice. Additionally, the same metric could be used to study tumor heterogeneity, e.g. by analyzing the distribution of active tumor fronts within the sample. More generally, our work could be extended into other areas of Computational Pathology beyond mitosis detection in breast tissue by: (1) adopting the combination of slide restaining, automatic image analysis and registration to create large-scale training datasets where immunohistochemistry is the reference standard; and (2) validating the proposed stain augmentation strategy in other applications that analyze H&E WSIs.

2.8 Acknowledgment

This study was financed by a grant from the Radboud Institute of Health Sciences (RIHS), Nijmegen, The Netherlands. The authors would like to thank Mitko Veta, organizer of the TUPAC Challenge, for evaluating our predictions in the test set of the TUPAC dataset; NVIDIA Corporation for donating a Titan X GPU card for our research; and the developers of Theano[79] and Lasagne[80], the open source tools that we used to run our deep learning experiments.

2.9 Appendix

2.9.1 Theory of color representation

The Lambert-Beer law describes the relation between the amount of light absorbed by a specimen and the amount of stain present on it:

$$\frac{I_i}{I_{0,i}} = \exp\left(-Ac_i\right) , \tag{2.1}$$

where I_i is the radiant flux emitted by the specimen, $I_{0,i}$ is the radiant flux received by the specimen, A is the amount of stain, c_i is the absorption factor, and subscript i indicates one of the RGB color channels.

Based on this law, we cannot establish a linear relationship between the relative absorption detected by a RGB camera ($I_i/I_{0,i}$) and the amount of stain present in a specimen. However, we can instead define the optical density (OD) per channel as follows:

$$OD_i = -\log\frac{I_i}{I_{0,i}} = -Ac_i . \tag{2.2}$$

Each OD vector describes a given stain in the OD-converted RGB color space. For example, measurements of a specimen stained with hematoxylin resulted in OD values of 0.18, 0.20 and 0.08 for each of the RGB channels, respectively[77].

By measuring the relative absorption for each RGB channel on slides stained with a single stain, Ruifrok et al.[77] quantified these OD vectors for hematoxylin, eosin and DAB (HED) stains. We can group these vectors into M, a 3 by 3 matrix representing a linear relationship between OD-converted pixels and the HED stain space. To achieve a correct balancing of the absorption factor for each stain, we divide each

OD vector in M by its total length.

Therefore, a particular set of OD-converted pixels y can be described as follows:

$$y = xM \ , \tag{2.3}$$

where x is a 1 by 3 vector representing the amount of stain (e.g. hematoxylin, eosin and DAB) per pixel, and M is the normalized OD matrix for the given combination of stains. Since we are interested in obtaining x, we only need to invert M:

$$x = yM^{-1} \ . \tag{2.4}$$

2.9.2 Color stain augmentation algorithm

Given an RGB image patch $P \in \mathbb{R}^{MxMx3}$ reshaped into $P \in \mathbb{R}^{Nx3}$ with N RGB pixels and the normalized OD matrix $M \in \mathbb{R}^{3x3}$ for hematoxylin, eosin and DAB, we apply equations 2.2 and 2.4 to transform the patch from RGB to HED color space as follows:

$$S = -\log{(P + \epsilon)}M^{-1} \ , \tag{2.5}$$

where $S \in \mathbb{R}^{Nx3}$ is the transformed patch in HED color space and ϵ is a positive bias to avoid numerical errors. We simulate alternative stain intensities by stochastically modifying each stain component:

$$S'_i = \alpha_i S_i + \beta_i \ , \tag{2.6}$$

where $S' \in \mathbb{R}^{Nx3}$ is the augmented patch in HED color space, subscript i represents each stain channel, α_i is drawn from a uniform distribution $U(1 - \sigma, 1 + \sigma)$, β_i is drawn from a uniform distribution $U(-\sigma, \sigma)$, and typically $\sigma = 0.05$.

To obtain an RGB representation of the augmented patch S', we invert the operations described in equation 2.5:

$$P' = \exp{(-S'M)} - \epsilon \ , \tag{2.7}$$

where $P' \in \mathbb{R}^{Nx3}$ is the augmented patch in RGB color space. Finally, we reshape P' into $P' \in \mathbb{R}^{MxMx3}$ to match the original shape of the patch.

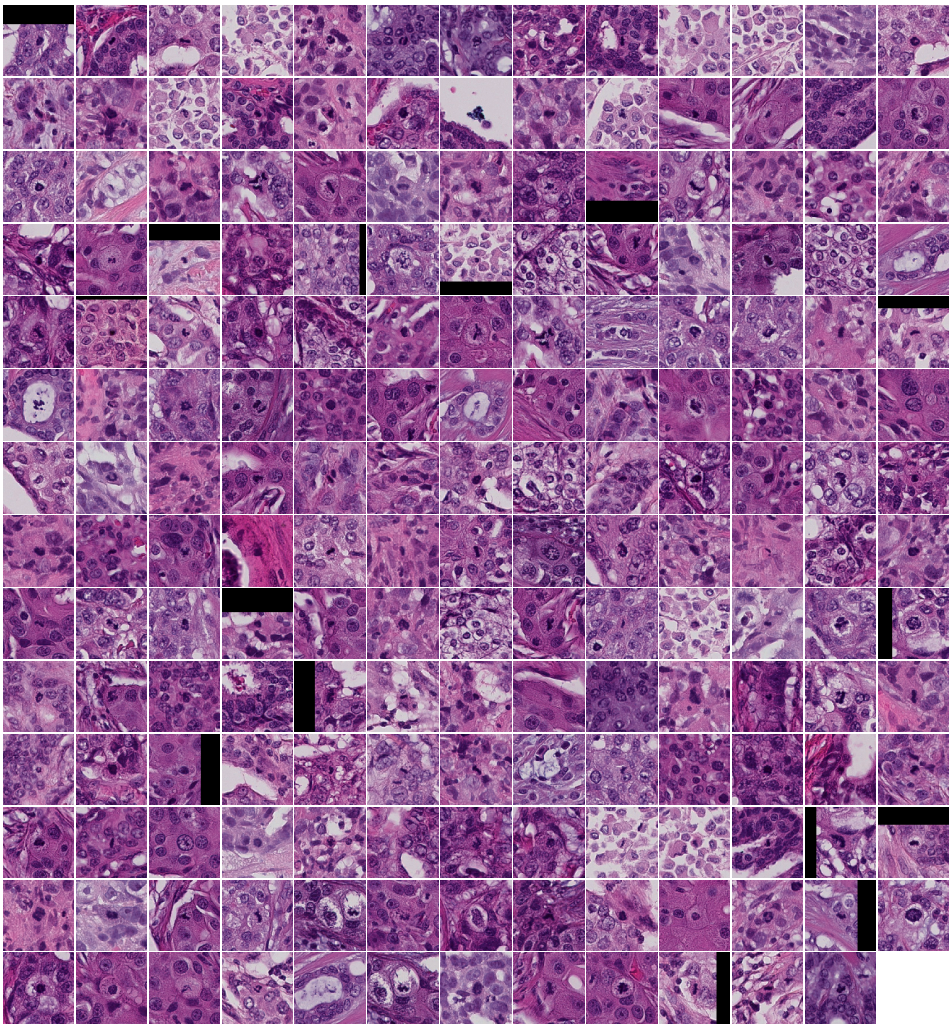

Figure 2.8: Mitosis detections in the *TUPAC-aux-test* dataset identified by *CNN5*. These patches were classified as containing a mitotic figure or not by a resident pathologist. The observer classified 128 out of 181 detections as true positives, resulting in a precision score of 0.707 for the automatic detector.

Color augmentation and normalization in computational pathology

3

Authors: David Tellez, Geert Litjens, Peter Bandi, Wouter Bulten, John-Melle Bokhorst, Francesco Ciompi, and Jeroen van der Laak

Original title: Quantifying the effects of data augmentation and stain color normalization in convolutional neural networks for computational pathology

Published in: Elsevier Medical Image Analysis (Volume: 58, Dec. 2019)

DOI URL: doi.org/10.1016/j.media.2019.101544

Abstract

Stain variation is a phenomenon observed when distinct pathology laboratories stain tissue slides that exhibit similar but not identical color appearance. Due to this color shift between laboratories, convolutional neural networks (CNNs) trained with images from one lab often underperform on unseen images from the other lab.

Several techniques have been proposed to reduce the generalization error, mainly grouped into two categories: stain color augmentation and stain color normalization. The former simulates a wide variety of realistic stain variations during training, producing stain-invariant CNNs. The latter aims to match training and test color distributions in order to reduce stain variation.

For the first time, we compared some of these techniques and quantified their effect on CNN classification performance using a heterogeneous dataset of hematoxylin and eosin histopathology images from 4 organs and 9 pathology laboratories. Additionally, we propose a novel unsupervised method to perform stain color normalization using a neural network.

Based on our experimental results, we provide practical guidelines on how to use stain color augmentation and stain color normalization in future computational pathology applications.

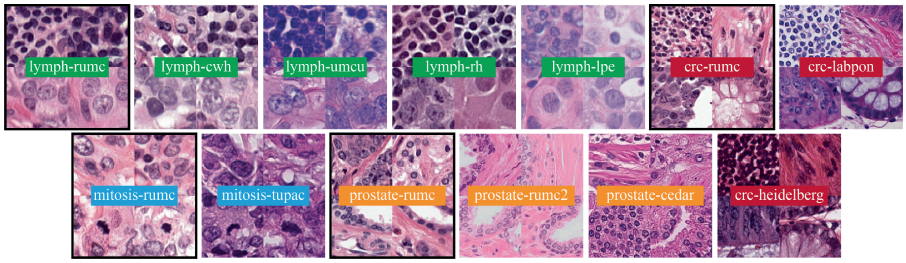

Figure 3.1: Example images from training and test datasets. Applications are indicated by colors and keywords: tumor detection in lymph nodes (*lymph*), colorectal cancer tissue classification (*crc*), mitosis detection (*mitosis*) and prostate epithelium detection (*prostate*). Training set images are indicated by the keyword *rumc* and black outline. The rest belong to test sets from other centers. Stain variation can be observed between training and test images.

3.1 Introduction

Computational pathology aims at developing machine learning based tools to automate and streamline the analysis of whole-slide images (WSI), i.e. high-definition images of histological tissue sections. These sections consist of thin slices of tissue that are stained with different dyes so that tissue architecture becomes visible under the microscope. In this study, we focus on hematoxylin and eosin (H&E), the most widely used staining worldwide. It highlights cell nuclei in blue color (hematoxylin), and cytoplasm, connective tissue and muscle in various shades of pink (eosin). The eventual color distribution of the WSI depends on multiple steps of the staining process, resulting in slightly different color distributions depending on the laboratory where the sections were processed, see Fig. 3.1 for examples of H&E stain variation. This inter-center stain variation hampers the performance of machine learning algorithms used for automatic WSI analysis. Algorithms that were trained with images originated from a single pathology laboratory often underperform when applied to images from a different center, including state-of-the-art methods based on convolutional neural networks (CNNs)[18,81–83]. Existing solutions to reduce the generalization error in this setting can be categorized into two groups: (1) *stain color augmentation*, and (2) *stain color normalization*.

3.1.1 Stain color augmentation

Stain color augmentation, and more generally data augmentation, has been proposed as a method to reduce CNN generalization error by simulating realistic variations of the training data. These artificial variations are hand-engineered to mimic the appearance of future test samples that deviate from the training manifold. Previous work on data augmentation for computational pathology has defined two main groups of augmentation techniques: (1) morphological and (2) color transformations[71,84]. Morphological augmentation spans from simple techniques such as 90-degree rotations, vertical and horizontal mirroring, or image scaling; to more advanced methods like elastic deformation[76], additive Gaussian noise, and Gaussian blurring. The common denominator among these transformations is the fact that only the morphology of the underlying image is modified and not the color appearance, e.g. Gaussian blurring simulates out of focus artifacts which is a common issue encountered with WSI scanners. Conversely, color augmentation leaves morphological features intact and focuses on simulating stain color variations instead. Common color augmentation techniques borrowed from Computer Vision include brightness, contrast and hue perturbations. Recently, researchers have proposed other approaches more tailored to mimic specific H&E stain variations, e.g. by perturbing the images directly in the H&E color space[84], or perturbing the principal components of the pixel values[85].

3.1.2 Stain color normalization

Stain color normalization reduces stain variation by matching the color distribution of the training and test images. Traditional approaches try to normalize the color space by estimating a color deconvolution matrix that allows identifying the underlying stains[68,86]. More recent methods use machine learning algorithms to detect certain morphological structures, e.g. cell nuclei, that are associated with certain stains, improving the result of the normalization process[69,70]. Deep generative models, i.e. variational autoencoders and generative adversarial networks[87,88], have been used to generate new image samples that match the template data manifold[89,90]. Moreover, color normalization has been formulated as a style transfer task where the style is defined as the color distribution produced by a particular lab[85]. However, despite their success and widespread adoption as a preprocessing tool in a variety of computational pathology applications[91–94], they are not always effective and can produce images with color distributions that deviate from the desired color template. In this study, we propose a novel unsupervised approach that

leverages the power of deep learning to solve the problem of stain normalization. We reformulate the problem of stain normalization as an image-to-image translation task and train a neural network to solve it. We do so by feeding the network with heavily augmented H&E images and training the model to reconstruct the original image without augmentation. By learning to remove this color variation, the network effectively learns to perform *stain color normalization* in unseen images whose color distribution deviates from that of the training set.

3.1.3 Multicenter evaluation

Despite the wide adoption of *stain color augmentation* and *stain color normalization* in the field of computational pathology, the effects on performance of these techniques have not been systematically evaluated. Existing literature focuses on particular applications, and does not quantify the relationship between these techniques and CNN performance[81,82,95,96]. In this study, we aim to overcome this limitation by comparing these techniques across four representative applications including multicenter data. We selected four patch-based classification tasks where a CNN was trained with data from a single center only, and evaluated in unseen data from multiple external pathology laboratories. We chose four relevant applications from the literature: (1) detecting the presence of mitotic figures in breast tissue[84]; (2) detecting the presence of tumor metastases in breast lymph node tissue[94]; (3) detecting the presence of epithelial cells in prostate tissue[97]; and (4) distinguishing among 9 tissue classes in colorectal cancer (CRC) tissue[98]. All test datasets presented a substantial and challenging stain color deviation from the training set, as can be seen in Fig. 3.1. We trained a series of CNN classifiers following an identical training protocol while varying the *stain color normalization* and *stain color augmentation* techniques used during training. This thorough evaluation allowed us to establish a ranking among the methods and measure relative performance improvements among them.

3.1.4 Contributions

Our contributions can be summarized as follows:

- We systematically evaluated several well-known *stain color augmentation* and *stain color normalization* algorithms in order to quantify their effects on CNN classification performance.

- We conducted the previous evaluation using data from a total of 9 different centers spanning 4 relevant classification tasks: mitosis detection, tumor metastasis detection in lymph nodes, prostate epithelium detection, and multiclass colorectal cancer tissue classification.

- We formulated the problem of *stain color normalization* as an unsupervised image-to-image translation task and trained a neural network to solve it.

This chapter is organized as follows. Sec. 3.2 and Sec. 3.3 describe the materials and methods thoroughly. Experimental results are explained in Sec. 3.4, followed by Sec. 3.5 and Sec. 3.6 where the discussion and final conclusion are stated.

3.2 Materials

We collected data from a variety of pathology laboratories for four different applications. In all cases, we used images from the Radboud University Medical Centre (Radboudumc or *rumc*) exclusively to train the models for each of the four classification tasks. Images from the remaining centers were used for testing purposes only. We considered RGB patches of 128x128 pixels extracted from annotated regions. Examples of these patches are shown in Fig. 3.1. The following sections describe each of the four classification tasks.

3.2.1 Mitotic figure detection

In this binary classification task, the goal was to accurately classify as positive samples those patches containing a mitotic figure in their center, i.e. a cell undergoing division. In order to train the classifier, we used 14 H&E WSIs from triple negative breast cancer patients, scanned at $0.25\,\mu\mathrm{m/pixel}$ resolution, with annotations of mitotic figures obtained as described in[84]. We split the slides into training (6), validation (4) and test (4), and extracted a total of 1M patches. We refer to this set as *mitosis-rumc*.

For the external dataset, we used publicly available data from the TUPAC Challenge[82], i.e. 50 cases of invasive breast cancer with manual annotations of mitotic figures scanned at $0.25\,\mu\mathrm{m/pixel}$ resolution. We extracted a total of 300K patches, and refer to this dataset as *mitosis-tupac*.

3.2.2 Tumor metastasis detection

The aim of this binary classification task was to identify patches containing metastatic tumor cells. We used publicly available WSIs from the Camelyon17 Challenge[94]. This cohort consisted of 50 exhaustively annotated H&E slides of breast lymph node resections from breast cancer patients from 5 different centers (10 slides per center), including Radboudumc. They were scanned at $0.25\,\mu\mathrm{m/pixel}$ resolution and the tumor metastases were manually delineated by experts.

We used the 10 WSIs from the Radboudumc to train the classifier, split into training (4), validation (3) and test (3), and extracted a total of 300K patches. We refer to this dataset as *lymph-rumc*. We used the remaining 40 WSIs as external test data, extracting a total of 1.2M patches, and assembling 4 different test sets (one for each center). We named them according to their center's name acronym: *lymph-umcu*, *lymph-cwh*, *lymph-rh* and *lymph-lpe*.

3.2.3 Prostate epithelium detection

The goal of this binary classification task was to identify patches containing epithelial cells in prostate tissue. We trained the classifier with 25 H&E WSIs of prostate resections from the Radboudumc scanned at $0.5\,\mu\mathrm{m/pixel}$ resolution, with annotations of epithelial tissue as described in[97]. We split this cohort into training (13), validation (6) and test (6), and extracted a total of 250K patches. We refer to it as *prostate-rumc*.

We used two test datasets for this task. First, we selected 10 H&E slides of prostate resections from the Radboudumc with different staining and scanning conditions, resulting in substantially different stain appearance (see *prostate-rumc2* in Fig. 3.1). This test set was manually annotated as described in[97] and named *prostate-rumc2*. We extracted 75K patches from these WSIs. Second, we used publicly available images from 20 H&E slides of prostatectomy specimens with manual annotations of epithelial tissue obtained as described in[97,99]. We extracted 65K patches from them and named the test set *prostate-cedar*.

3.2.4 Colorectal cancer tissue type classification

In this multiclass classification task, the goal was to distinguish among 9 different colorectal cancer (CRC) tissue classes, namely: 1) tumor, 2) stroma, 3) muscle, 4) lymphocytes, 5) healthy glands, 6) fat, 7) blood cells, 8) necrosis and debris, and 9) mucus. We used 54 H&E WSIs of colorectal carcinoma tissue from the Radboudumc scanned at $0.5\,\mu\text{m/pixel}$ resolution to train the classifier, with manual annotations of the 9 different tissue classes. We split this cohort into training (24), validation (15) and test (15), extracted a total of 450K patches, and named it *crc-rumc*.

We used two external datasets for this task. First, a set of 74 H&E WSIs from rectal carcinoma patients with annotations of the same 9 classes, as described in[98]. We extracted 35K patches and refer to this dataset as *crc-labpon*. Second, we used a publicly available set of H&E image patches from colorectal carcinoma patients[100]. Annotations for 6 tissue types were available: 1) tumor, 2) stroma, 3) lymph, 4) healthy glands, 5) fat, and 6) blood cells, debris and mucus. We extracted 4K patches in total, and refer to this dataset as *crc-heidelberg*.

3.2.5 Multi-organ dataset

For the purpose of training a network to solve the problem of *stain color normalization*, we created an auxiliary dataset by aggregating patches from *mitosis-rumc*, *lymph-rumc*, *prostate-rumc* and *crc-rumc* in a randomized and balanced manner. We discarded all labels since they were not needed for this purpose. We preserved a total of 500K patches for this set and called it the *multi-organ* dataset.

3.3 Methods

In this study, we evaluated the effect in classification performance of several methods for *stain color augmentation* and *stain color normalization*. This section describes these methods.

3.3.1 Stain color augmentation

We assume a homogeneous stain color distribution ϕ_{train} for the training images and a more varied color distribution ϕ_{test} for the test images. Note that it is challenging

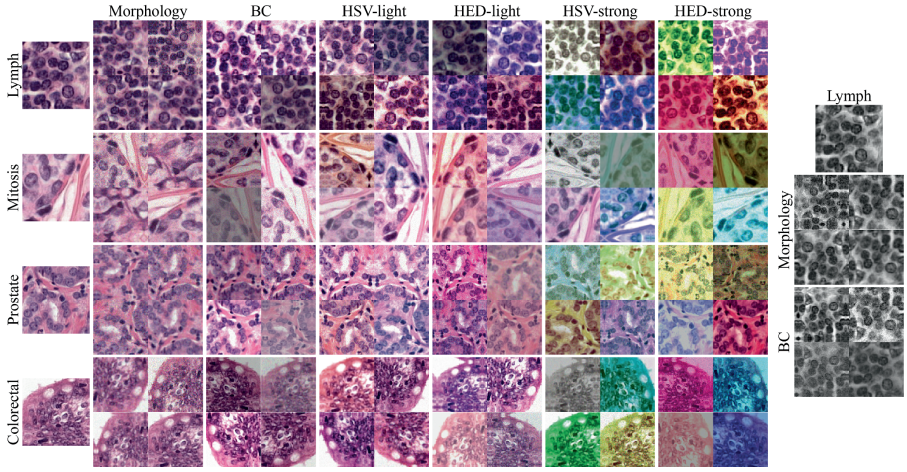

Figure 3.2: Summary of the data augmentation techniques and datasets used in this study, organized in columns and rows respectively. Patches on the leftmost column depict the original input images and the rest of patches are augmented versions of them. Augmentations performed in the *grayscale* color space are depicted on the right for one sample dataset only. *Basic* augmentation is included in all cases.

for a classification model trained solely with ϕ_{train} to generalize well to ϕ_{test} due to potential stain differences among sets. To solve this problem, stain color augmentation defines a preprocessing function f that transforms images of the training set to present an alternative and more diverse color distribution ϕ_{augment}:

$$\phi_{\text{train}} \xrightarrow{f} \phi_{\text{augment}} \tag{3.1}$$

on the condition that:

$$(\phi_{\text{augment}} \supseteq \phi_{\text{train}}) \wedge (\phi_{\text{augment}} \supseteq \phi_{\text{test}}) \tag{3.2}$$

In practice, heavy data augmentation is used to satisfy Eq. 3.2. In order to simplify our experimental setup, we grouped several data augmentation techniques into the following categories attending to the nature of the image transformations. Examples of the resulting augmented images are shown in Fig. 3.2.

Basic. This group included 90 degree rotations, and vertical and horizontal mirroring.

Morphology. We extended *basic* with several transformations that simulate morphological perturbations, i.e. alterations in shape, texture or size of the imaged tissue structures, including scanning artifacts. We included *basic* augmentation, scaling, elastic deformation[76], additive Gaussian noise (perturbing the signal-to-noise ratio), and Gaussian blurring (simulating out-of-focus artifacts).

Brightness & contrast (BC). We extended *morphology* with random brightness and contrast image perturbations[101].

Hue-Saturation-Value (HSV). We extended the *BC* augmentation by randomly shifting the hue and saturation channels in the HSV color space[102]. This transformation produced substantially different color distributions when applied to the training images. We tested two configurations depending on the observed color variation strength, called *HSV-light* and *HSV-strong*.

Hematoxylin-Eosin-DAB (HED). We extended the *BC* augmentation with a color variation routine specifically designed for H&E images[84]. This method followed three steps. First, it disentangled the hematoxylin and eosin color channels by means of color deconvolution using a fixed matrix. Second, it perturbed the hematoxylin and eosin stains independently. Third, it transformed the resulting stains into regular RGB color space. We tested two configurations depending on the observed color variation strength, called *HED-light* and *HED-strong*.

During training, we selected the value of the augmentation hyper-parameters randomly within certain ranges to achieve stain variation. We tuned all ranges manually via visual examination. In particular, we used a scaling factor between $[0.8, 1.2]$, elastic deformation parameters $\alpha \in [80, 120]$ and $\sigma \in [9.0, 11.0]$, additive Gaussian noise with $\sigma \in [0, 0.1]$, Gaussian blurring with $\sigma \in [0, 0.1]$, brightness intensity ratio between $[0.65, 1.35]$, and contrast intensity ratio between $[0.5, 1.5]$. For *HSV-light* and *HSV-strong*, we used hue and saturation intensity ratios between $[-0.1, 0.1]$ and $[-1, 1]$, respectively. For *HED-light* and *HED-strong*, we used intensity ratios between $[-0.05, 0.05]$ and $[-0.2, 0.2]$, respectively, for all HED channels.

3.3.2 Stain color normalization

Stain color normalization reduces color variation by transforming the color distribution of training and test images, i.e. ϕ_{train} and ϕ_{test}, to that of a template ϕ_{normal}. It

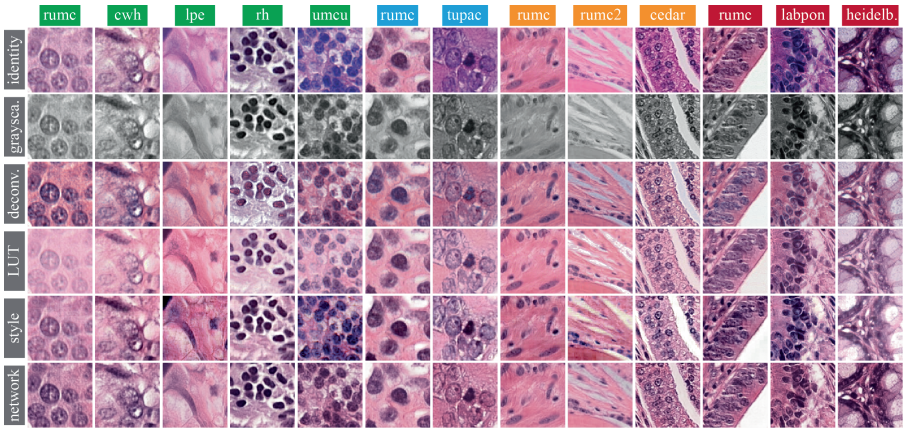

Figure 3.3: Visual comparison of the stain color normalization techniques used in this study. Rows correspond to the different tested techniques and columns to datasets, with green for *lymph*, blue for *mitosis*, yellow for *prostate* and red for *colorectal*.

performs such transformation using a normalization function g that maps any given color distribution to the template one:

$$(\phi_{\text{train}} \xrightarrow{g} \phi_{\text{normal}}) \wedge (\phi_{\text{test}} \xrightarrow{g} \phi_{\text{normal}}) \tag{3.3}$$

By matching ϕ_{train} and ϕ_{test}, the problem of stain variance vanishes and the model no longer requires to generalize to unseen stains in order to perform well. We evaluated several methods that implement g (see Fig. 3.3), and propose a novel technique based on neural networks.

Identity. We performed no transformation on the input patches, serving as a baseline method for the rest of techniques.

Grayscale. In this case, g transformed images from RGB to grayscale space, removing most of the color information present in the patches. We hypothesized that this color information is redundant since most of the signal in H&E images is present in morphological and structural patterns, e.g. the presence of a certain type of cell.

Deconv-based. We followed the color deconvolution approach proposed by [68]. This method assumes that the hematoxylin and eosin stains are linearly separable in the optical density (OD) color space, as opposed to RGB space. This method finds the two largest singular value directions using singular value decomposition, and

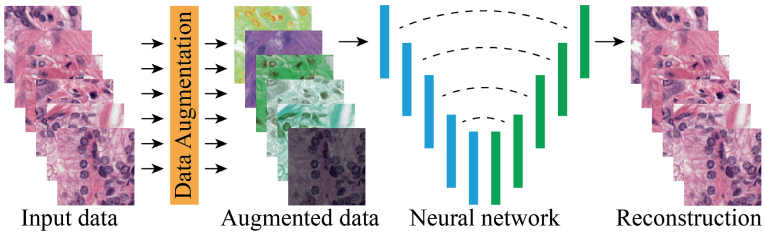

Figure 3.4: Network-based stain color normalization. From left to right: patches from the training set are transformed with heavy color augmentation and fed to a neural network. This network is trained to reconstruct the original appearance of the input images by removing color augmentation, effectively learning how to perform stain color normalization.

projects the OD pixel values onto this plane. This procedure allows to identify the underlying hematoxylin and eosin stain vectors, and use them to perform color deconvolution on a given image to decompose the RGB image into its normalized hematoxylin and eosin components.

LUT-based. We implemented an approach that uses tissue morphology to perform stain color normalization[70]. This popular method has been used by numerous researchers in recent public challenges[94,103]. It detects cell nuclei in order to precisely characterize the H&E chromatic distribution and density histogram for a given WSI. First, it does so for a given template WSI, e.g. an image from the training set, and a target WSI. Second, the color distributions of the template and target WSIs are matched, and the color correspondence is stored in a look-up table (LUT). Finally, this LUT is used to normalize the color of the target WSI.

Style-based. A recent study[85] proposed to use a neural network to perform stain color normalization based on the idea of style transfer. They transform the color distribution of RGB images by using feature-aware normalization, a mechanism that shifts and scales intermediate feature maps based on features extracted from the input image. This feature extractor is an ImageNet[57] pre-trained network, while the rest of the model is trained to reconstruct PCA-augmented histopathology images. We used the authors' implementation of the method and retrained the model using images from the *multi-organ* dataset.

Network-based. We developed a novel approach to perform stain color normalization based on unsupervised learning and neural networks (see Fig. 3.4). We parameterized the normalization function g with a neural network G and trained it

end-to-end to remove the effects of data augmentation. Even though it is not possible to invert the many-to-many augmentation function f, we can learn a partial many-to-one function that maps any arbitrary color distribution ϕ_{augment} to a template distribution ϕ_{normal}:

$$\phi_{\text{augment}} \xrightarrow{G} \phi_{\text{normal}} \tag{3.4}$$

Since G can normalize ϕ_{augment} (Eq. 3.4), and ϕ_{augment} is a superset of ϕ_{train} and ϕ_{test} (Eq. 3.2), we conclude that G can effectively normalize ϕ_{train} and ϕ_{test} (Eq. 3.2).

We trained G to perform image-to-image translation using the *multi-organ* dataset. During training, images were heavily augmented and fed to the network. The model was tasked with reconstructing the images with their original appearance, before augmentation. We used a special configuration of the *HSV* augmentation where we kept the color transformation only, i.e. did not include *basic*, *morphology* and *BC*. We used the maximum intensity for the transformation hyper-parameters, i.e. hue, saturation and value channel ratios between $[-1, 1]$. The strength of this augmentation resulted in images with drastically different color distributions, sometimes compressing all color information into grayscale. In order to invert this complex augmentation, we hypothesized that the network learned to associate certain tissue structures with their usual color appearance.

We used an architecture inspired by U-Net[104], with a downward path of 5 layers of strided convolutions[78] with 32, 64, 128, 256 and 512 3x3 filters, stride of 2, batch normalization (BN)[40] and leaky-ReLU activation (LRA)[105]. The upward path consisted of 5 upsampling layers, each one composed of a pair of nearest-neighbor upsampling and a convolutional operation[106], with 256, 128, 64, 32 and 3 3x3 filters, BN and LRA; except for the final convolutional layer that did not have BN and used the hyperbolic tangent (tanh) as activation function. We used long skip connections to ease the synthesis upward path[104], and applied L2 regularization with a factor of 1×10^{-6}.

We minimized the mean squared error (MSE) loss using stochastic gradient descent with Adam optimization[39] and 64-sample mini-batch, decreasing the learning rate by a factor of 10 starting from 1×10^{-2} every time the validation loss stopped improving for 4 consecutive epochs until 1×10^{-5}. Finally, we selected the weights corresponding to the model with the lowest validation loss during training.

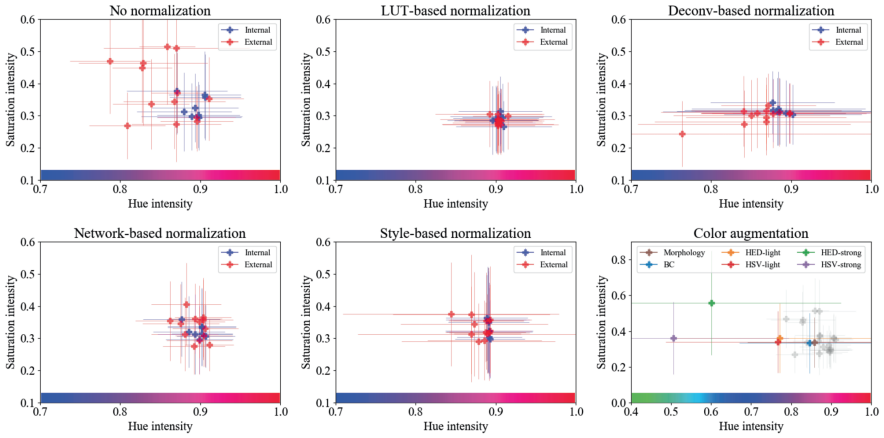

Figure 3.5: Constellations of internal and external datasets analyzed in this work. Each data point represents the mean and standard deviation pixel intensity of all image patches in a particular dataset in the HSV color space (hue and saturation in the x and y axis, respectively). Note how normalization methods tend to cluster the color distribution of the datasets, whereas color augmentation does the opposite. Color augmentation plot (bottom-right): patches from internal images are transformed with different color augmentation methods (grey points representing the original internal and external datasets are shown as reference).

Convergence to average solutions is a known effect with bottleneck architectures trained with MSE loss. Note, however, that our network-based normalization architecture includes long skip connections between the downward and the upward paths. These skip connections allow the model to copy spatial structures from the input images to the output images with ease, and utilize the rest of the model to modify style-related color features. Since there is no bottleneck effect, i.e., the model has all the information necessary to reconstruct the input image, image reconstructions are highly accurate and do not show any blurriness in practice.

3.3.3 Color analysis

In order to understand how *stain color augmentation* and *stain color normalization* influenced the color differences between internal (*rumc*) and external datasets (rest), we analyzed the image patches in the HSV color space. We measured the mean and standard deviation pixel intensity along the hue and saturation dimensions, and plotted the results in a 2D plane, comparing images processed with the color normalization

and augmentation techniques analyzed in this work (see Fig. 3.5). We confirmed the clustering effect of normalization algorithms, and the scattering effect of augmentation methods.

3.3.4 CNN Classifiers

In order to measure the effect of *stain color augmentation* and *stain color normalization*, we trained a series of identical CNN classifiers to perform patch classification using different combinations of these techniques. For training and validation purposes, we used the *rumc* datasets described in Sec. 3.2.

The architecture of such CNN classifiers consisted of 9 layers of strided convolutions with 32, 64, 64, 128, 128, 256, 256, 512 and 512 3x3 filters, stride of 2 in the even layers, BN and LRA; followed by global average pooling; 50% dropout; a dense layer with 512 units, BN and LRA; and a linear dense layer with either 2 or 9 units depending on the classification task, followed by a softmax. We applied L2 regularization with a factor of 1×10^{-6}.

We minimized the cross-entropy loss using stochastic gradient descent with Adam optimization and 64-sample class-balanced mini-batch, decreasing the learning rate by a factor of 10 starting from 1×10^{-2} every time the validation loss stopped improving for 4 consecutive epochs until 1×10^{-5}. Finally, we selected the weights corresponding to the model with the lowest validation loss during training.

3.4 Experimental results

We conducted a series of experiments in order to quantify the impact in performance of the different *stain color augmentation* and *stain color normalization* methods introduced in the previous section across four different classification tasks. We trained a CNN classifier for each combination of organ, color normalization and data augmentation method under consideration. In the case of *grayscale* normalization, we only tested *basic*, *morphology* and *BC* augmentation techniques. We conducted 152 different experiments, repeating each 5 times using different random initialization for the network parameters, accounting for a total of 760 trained CNN classifiers.

Table 3.1: Experimental results ranking *stain color augmentation* and *stain color normalization* methods. Values correspond to AUC scores, except for the last column, averaged across 5 repetitions with standard deviation shown between parenthesis. Each column represents a different external test dataset, with the last column *Ranking* indicating the position of each method within the global benchmark, computed as described in Sec. 3.4.1. Normalization methods: *Network* (our proposal), *Style*[85], *LUT*[70], and *Deconvolution*[68].

Normalization	Augmentation	lymph-cwh	lymph-lpe	lymph-rh	lymph-umcu	mitosis-tupac	prostate-rumc2	prostate-cedar	crc-labpon	crc-heidelberg	Ranking
Identity	HED-light	0.952(0.004)	0.976(0.001)	0.946(0.009)	0.968(0.004)	0.996(0.001)	0.957(0.001)	0.879(0.011)	0.973(0.002)	0.895(0.002)	1.2(0.4)
Style	HED-light	0.961(0.002)	0.953(0.004)	0.952(0.001)	0.972(0.004)	0.991(0.003)	0.925(0.003)	0.879(0.006)	0.975(0.001)	0.917(0.003)	2.8(0.7)
Network	HSV-light	0.946(0.006)	0.962(0.001)	0.941(0.002)	0.965(0.004)	0.992(0.001)	0.957(0.000)	0.872(0.013)	0.980(0.001)	0.900(0.003)	3.9(1.9)
Network	HED-light	0.949(0.005)	0.968(0.001)	0.942(0.002)	0.963(0.004)	0.989(0.003)	0.958(0.001)	0.862(0.011)	0.980(0.001)	0.906(0.003)	4.1(1.6)
Identity	HSV-strong	0.955(0.003)	0.965(0.004)	0.929(0.002)	0.973(0.003)	0.988(0.003)	0.945(0.009)	0.886(0.005)	0.977(0.001)	0.902(0.003)	4.7(1.7)
Network	HSV-strong	0.953(0.002)	0.964(0.003)	0.946(0.002)	0.964(0.005)	0.991(0.003)	0.951(0.002)	0.852(0.006)	0.975(0.002)	0.894(0.005)	6.6(0.9)
Network	HED-strong	0.956(0.003)	0.959(0.002)	0.940(0.003)	0.965(0.004)	0.985(0.005)	0.943(0.003)	0.861(0.009)	0.974(0.002)	0.916(0.003)	7.9(1.9)
Identity	HED-strong	0.950(0.005)	0.959(0.005)	0.936(0.007)	0.957(0.007)	0.992(0.002)	0.945(0.003)	0.872(0.005)	0.967(0.003)	0.920(0.005)	8.1(2.8)
Style	HSV-strong	0.953(0.004)	0.956(0.004)	0.940(0.003)	0.959(0.007)	0.986(0.004)	0.932(0.005)	0.878(0.003)	0.976(0.001)	0.917(0.004)	9.0(2.6)
Style	HSV-light	0.940(0.011)	0.960(0.004)	0.944(0.007)	0.926(0.012)	0.992(0.001)	0.958(0.002)	0.852(0.008)	0.974(0.001)	0.921(0.003)	9.4(3.6)
Style	HED-strong	0.955(0.002)	0.949(0.004)	0.936(0.003)	0.954(0.005)	0.982(0.005)	0.942(0.002)	0.884(0.004)	0.975(0.000)	0.925(0.003)	9.9(1.2)
Grayscale	BC	0.956(0.003)	0.962(0.003)	0.935(0.005)	0.961(0.002)	0.989(0.002)	0.939(0.004)	0.851(0.002)	0.972(0.000)	0.884(0.003)	12.2(1.2)
Deconvolution	HSV-strong	0.955(0.003)	0.936(0.008)	0.941(0.004)	0.943(0.009)	0.991(0.001)	0.865(0.010)	0.867(0.004)	0.961(0.002)	0.928(0.001)	13.9(1.9)
LUT	HED-strong	0.934(0.006)	0.941(0.006)	0.925(0.006)	0.963(0.005)	0.989(0.002)	0.945(0.002)	0.871(0.005)	0.956(0.002)	0.945(0.001)	14.0(2.2)
Deconvolution	HED-strong	0.942(0.003)	0.962(0.003)	0.897(0.006)	0.967(0.003)	0.993(0.002)	0.827(0.018)	0.853(0.006)	0.969(0.001)	0.927(0.002)	14.4(1.2)
LUT	HSV-strong	0.923(0.009)	0.939(0.003)	0.928(0.005)	0.947(0.008)	0.987(0.002)	0.949(0.003)	0.862(0.007)	0.962(0.002)	0.940(0.002)	17.0(2.0)
Network	BC	0.944(0.003)	0.950(0.003)	0.903(0.003)	0.934(0.006)	0.983(0.005)	0.953(0.003)	0.869(0.009)	0.981(0.001)	0.881(0.005)	17.4(1.5)
Identity	HSV-light	0.888(0.013)	0.951(0.009)	0.942(0.004)	0.930(0.023)	0.962(0.015)	0.949(0.001)	0.905(0.005)	0.976(0.000)	0.894(0.003)	17.4(2.9)
LUT	HED-light	0.914(0.011)	0.926(0.011)	0.923(0.006)	0.932(0.019)	0.993(0.001)	0.948(0.003)	0.852(0.021)	0.966(0.003)	0.940(0.002)	17.9(2.2)
LUT	HSV-light	0.894(0.006)	0.936(0.006)	0.921(0.003)	0.942(0.007)	0.987(0.002)	0.951(0.002)	0.860(0.010)	0.971(0.002)	0.945(0.002)	19.2(1.2)
LUT	BC	0.925(0.025)	0.948(0.027)	0.853(0.016)	0.790(0.061)	0.985(0.004)	0.951(0.004)	0.848(0.018)	0.973(0.003)	0.924(0.005)	21.3(3.3)
Style	BC	0.949(0.005)	0.858(0.031)	0.938(0.001)	0.411(0.065)	0.987(0.004)	0.949(0.006)	0.764(0.047)	0.946(0.002)	0.903(0.005)	23.3(2.2)
Deconvolution	HSV-light	0.942(0.004)	0.930(0.009)	0.913(0.023)	0.961(0.002)	0.982(0.005)	0.850(0.019)	0.840(0.009)	0.958(0.006)	0.917(0.002)	23.5(1.0)
Network	Basic	0.944(0.003)	0.954(0.007)	0.887(0.010)	0.959(0.004)	0.969(0.005)	0.905(0.006)	0.815(0.019)	0.977(0.002)	0.855(0.006)	23.6(1.4)
Network	Morphology	0.939(0.010)	0.949(0.006)	0.890(0.012)	0.950(0.009)	0.980(0.006)	0.913(0.011)	0.823(0.022)	0.977(0.001)	0.868(0.002)	23.9(1.1)
Deconvolution	HED-light	0.930(0.005)	0.912(0.015)	0.916(0.005)	0.948(0.006)	0.982(0.002)	0.816(0.011)	0.834(0.004)	0.970(0.003)	0.927(0.005)	25.4(2.3)
Deconvolution	Morphology	0.951(0.003)	0.938(0.006)	0.849(0.021)	0.951(0.008)	0.993(0.002)	0.754(0.027)	0.749(0.037)	0.903(0.008)	0.865(0.015)	27.7(0.6)
Grayscale	Morphology	0.943(0.010)	0.820(0.021)	0.922(0.005)	0.941(0.011)	0.991(0.006)	0.910(0.009)	0.816(0.005)	0.929(0.006)	0.813(0.009)	27.7(1.2)
Style	Morphology	0.935(0.011)	0.725(0.082)	0.934(0.002)	0.361(0.113)	0.992(0.004)	0.918(0.006)	0.754(0.006)	0.873(0.010)	0.890(0.006)	28.6(2.4)
Grayscale	Basic	0.940(0.007)	0.692(0.064)	0.926(0.010)	0.938(0.019)	0.992(0.001)	0.882(0.008)	0.661(0.039)	0.934(0.002)	0.798(0.006)	30.0(0.6)
Deconvolution	BC	0.942(0.004)	0.896(0.008)	0.682(0.044)	0.949(0.005)	0.989(0.006)	0.794(0.021)	0.792(0.028)	0.930(0.007)	0.872(0.004)	30.4(1.2)
LUT	Morphology	0.898(0.007)	0.920(0.007)	0.801(0.021)	0.874(0.025)	0.969(0.008)	0.895(0.013)	0.803(0.007)	0.939(0.006)	0.906(0.006)	32.6(1.4)
Deconvolution	Basic	0.919(0.015)	0.896(0.038)	0.810(0.081)	0.902(0.026)	0.993(0.001)	0.753(0.006)	0.791(0.009)	0.903(0.003)	0.836(0.008)	32.8(0.7)
Style	Basic	0.918(0.002)	0.334(0.133)	0.926(0.004)	0.124(0.041)	0.991(0.003)	0.865(0.025)	0.723(0.024)	0.863(0.020)	0.857(0.010)	33.6(1.0)
LUT	Basic	0.908(0.010)	0.894(0.030)	0.809(0.022)	0.772(0.072)	0.951(0.009)	0.906(0.011)	0.741(0.018)	0.930(0.014)	0.890(0.013)	34.6(0.8)
Identity	BC	0.899(0.006)	0.634(0.100)	0.741(0.016)	0.177(0.047)	0.906(0.034)	0.936(0.006)	0.704(0.060)	0.684(0.009)	0.761(0.012)	36.2(0.4)
Identity	Morphology	0.811(0.026)	0.671(0.099)	0.673(0.027)	0.214(0.174)	0.986(0.006)	0.374(0.191)	0.602(0.023)	0.569(0.028)	0.720(0.009)	37.2(0.7)
Identity	Basic	0.811(0.009)	0.563(0.309)	0.790(0.047)	0.406(0.375)	0.965(0.009)	0.631(0.178)	0.624(0.053)	0.556(0.057)	0.701(0.028)	37.6(0.5)

3.4.1 Evaluation

We evaluated the area under the receiver-operating characteristic curve (AUC) of each CNN in each external test set. In the case of multiclass classification, we considered the unweighted average, i.e. we calculated the individual AUC per label (one-vs-all) and averaged the resulting values. We reported the mean and standard deviation of the resulting AUC for each experiment across five repetitions in Tab. 3.1.

In order to establish a global ranking among methods, shown in the rightmost column in Tab. 3.1, we performed the following calculation. We converted the AUC scores into ranking scores per test set column, and averaged these scores along the dataset dimension to obtain a global ranking score per method. Note that we performed an average across ranking scores, rather than AUC scores, following established procedures[107]. Data in Tab. 3.1 and the raw AUC scores are provided in machine-readable format as Supplementary Material to this article.

3.4.2 Effects of stain color augmentation

Results in Tab. 3.1 show that *stain color augmentation* was crucial to obtain top classification performance, regardless of the *stain color normalization* technique used (see top-10 methods). Moreover, note that including color augmentation, either *HSV* or *HED*, was key to obtaining top performance since using *BC* augmentation alone produced mediocre results. We did not find, however, any substantial performance difference between using *HED* or *HSV* color augmentation. Similarly, we found that *strong* and *light* color augmentations achieved similar performance, with a slight advantage towards *light*. Heavy augmentation is known to reduce performance on images similar to those in the training set. However, we found less than 1% average performance reduction on the internal test set across organs. Regarding non-color augmentation, i.e. *basic*, *morphology* and *BC*, *BC* obtained the best results across almost all *stain color normalization* setups, followed by *morphology* and *basic* augmentation, as expected.

3.4.3 Effects of stain color normalization

According to results in Tab. 3.1, overall top performance was achieved without the use of color normalization. This piece of evidence suggests that color normalization is not a necessary condition to achieve high classification performance in histopathol-

ogy images. However, we observed that color normalization generally produced classifiers that were more robust to different color augmentation techniques, e.g., *Identity* normalization performance diminished with *HSV-light* augmentation whereas *Network* normalization exhibited a high performance regardless of the color augmentation used.

We did not find any substantial performance difference between neural network based normalization algorithms, *Network* and *Style*. Nevertheless, we observed that none of the classical approaches, *LUT* or *Deconvolution*, surpassed the performance of *Grayscale*. We hypothesize that these classical normalization methods can hide certain useful features from the images, resulting in added input noise that can affect classification performance.

Additionally, we measured the extra time required to normalize a regular whole-slide image composed of 50000×50000 RGB pixels. We found *LUT-based* to be the fastest taking $21.8\,\mathrm{min}$, followed closely by *network-based* with $26.0\,\mathrm{min}$, and the slower *deconv-based* and *style-based* taking $111.2\,\mathrm{min}$ and $217.8\,\mathrm{min}$, respectively, excluding I/O delays.

3.5 Discussion

Our experimental results indicate that *stain color augmentation* improved classification performance drastically by increasing the CNN's ability to generalize to unseen stain variations. This was true for most of the experiments regardless of the type of *stain color normalization* technique used. Moreover, we found *HSV* and *HED* color transformations to be the key ingredients to improve performance since removing them, i.e. using *BC* augmentation, yielded a lower AUC under all circumstances; suggesting that inter-lab stain differences were mainly caused by color variations rather than morphological features. Remarkably, we observed hardly any performance difference between *HSV* or *HED*, and *strong* or *light* variation intensity.

Based on these observations, we concluded that CNNs are mostly insensitive to the type and intensity of the color augmentation used in this setup, as long as one of the methods is used. However, CNNs trained with simpler *stain color normalization* techniques exhibited more sensitivity to the intensity of color augmentation, i.e. they required a stronger augmentation in order to perform well. Finally, the fact that experiments with *grayscale* images achieved mediocre performance was an indication

that color provided useful information to the model. The worst performance was achieved with *morphology* and *identity* configurations, which was an indication that color information can act as noise when no augmentation is used, increasing overfitting and generalization error due to stain variation.

Regarding *stain color normalization*, we found that the best performing method did not use any normalization. This result challenged the common assumption that color normalization is a necessary step to achieve top classification performance in the histopathology setting; especially considering that color normalization added a computational overhead that can substantially reduce the overall classification speed. Neural network based methods, both *Network* and *Style*, achieved similar high performance on the benchmark, supporting the idea of reformulating the problem of *stain color normalization* as an image-to-image translation task.

Furthermore, we observed that all *stain color normalization* techniques obtained a poor performance when no color augmentation was used (below that of *Grayscale* with *BC*). We hypothesize that even in the case of excellent stain normalization, color information can serve as a source of overfitting, worsening with suboptimal normalization. We concluded that using the *stain color normalization* methods evaluated in this study without proper *stain color augmentation* is insufficient to reduce the generalization error caused by stain variation and results in poor model performance.

Due to computational constraints, we limited the type and number of experiments performed in this study to patch-based classification tasks, ignoring other modalities such as segmentation, instance detection or WSI classification. However, we believe this limitation to have little impact in the conclusions of this study since the problem of generalization error has identical causes and effects in other modalities. In order to reduce the number of experiments, we avoided quantifying the impact of individual augmentation techniques, e.g. scaling augmentation alone, but grouped them into categories instead. Similarly, we limited the hyper-parameters' ranges to certain set of values, e.g. *light* or *strong* stain augmentation intensity. Nevertheless, according to the experimental results, we believe that testing a wider range of hyper-parameter values would not alter the main conclusions of this study.

3.6 Conclusion

For the first time, we quantified the effect of *stain color augmentation* and *stain color normalization* in classification performance across four relevant computational pathology applications using data from 9 different centers. Based on our empirical evaluation, we found that any type of *stain color augmentation*, i.e. *HSV* or *HED* transformation, should always be used. In addition, color augmentation can be combined with neural network based *stain color normalization* to achieve a more robust classification performance. In setups with reduced computational resources, color normalization could be omitted, resulting in a negligible performance reduction and a substantial improvement in processing speed. Finally, we recommend tuning the intensity of the color augmentation to *light* or *strong* in case color normalization is *enabled* or *disabled*, respectively.

3.7 Acknowledgment

This study was supported by a Junior Researcher grant from the Radboud Institute of Health Sciences (RIHS), Nijmegen, The Netherlands; a grant from the Dutch Cancer Society (KUN 2015-7970); and another grant from the Dutch Cancer Society and the Alpe d'HuZes fund (KUN 2014-7032); this project has also received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825292.

The authors would like to thank Dr. Babak Ehteshami Bejnordi for providing the code for the *LUT-based* stain color normalization algorithm; and Canisius-Wilhelmina Ziekenhuis Nijmegen, Laboratorium Pathologie Oost Nederland, University Medical Center Utrecht, and Rijnstate Hospital Arnhem for kindly providing tissue sections for this study.

Neural image compression for gigapixel histopathology image analysis

4

Authors: David Tellez, Geert Litjens, Jeroen van der Laak, and Francesco Ciompi

Original title: Neural Image Compression for Gigapixel Histopathology Image Analysis

Published in: IEEE Transactions on Pattern Analysis and Machine Intelligence (Volume: 43, Issue: 2, Feb. 2021)

DOI URL: doi.org/10.1109/TPAMI.2019.2936841

Abstract

We propose Neural Image Compression (NIC), a two-step method to build convolutional neural networks for gigapixel image analysis solely using weak image-level labels.

First, gigapixel images are compressed using a neural network trained in an unsupervised fashion, retaining high-level information while suppressing pixel-level noise. Second, a convolutional neural network (CNN) is trained on these compressed image representations to predict image-level labels, avoiding the need for fine-grained manual annotations.

We compared several encoding strategies, namely reconstruction error minimization, contrastive training and adversarial feature learning, and evaluated NIC on a synthetic task and two public histopathology datasets.

We found that NIC can exploit visual cues associated with image-level labels successfully, integrating both global and local visual information. Furthermore, we visualized the regions of the input gigapixel images where the CNN attended to, and confirmed that they overlapped with annotations from human experts.

4.1 Introduction

Gigapixel images are three-dimensional arrays composed of more than 1 billion pixels; these are common in fields like Computational Pathology[28] and Remote Sensing[108], and are often associated with labels at image level. The fundamental challenge of gigapixel image analysis with weak image-level labels resides in the low signal-to-noise ratio present in these images. Typically, the signal consists of a subtle combination of high- and low-level patterns that are related to the image-level label, while most of the pixels behave as distracting noise. Furthermore, the nature and spatial distribution of the signal are both unknown, often referred to as the *what* and the *where* problems, respectively.

4.1.1 The *what* and the *where* problems

Researchers have addressed the challenge of gigapixel image analysis by making different assumptions about the signal, simplifying either the *what* or the *where* problem.

The most widespread simplification assumes that the signal is fully recognizable at a low level of abstraction, i.e., the image-level label has a patch-level representation. This simplification addresses the *what* problem by decomposing the gigapixel image into a set of patches that can be independently annotated. Typically, these patches are manually annotated to perform automatic detection or segmentation using a neural network, relegating the task of performing image-level prediction to a rule-based decision model about the patch-level predictions[28,82,83,109]. This assumption is not valid for image-level labels that do not have a known patch-level representation. Furthermore, patch-level annotation in gigapixel images is a tedious, time consuming and error-prone process, and limits what machine learning models can learn to the knowledge of human annotators.

Other researchers have assumed that the signal can exist at a low level of abstraction, but it is then not fully recognizable, i.e., the image-level label has a patch-level representation that is unknown to human annotators. Furthermore, the mere presence of these patches is enough evidence to make a prediction at the image level, ignoring the spatial arrangement between patches, thus solving the *where* problem. Making this assumption falls into the multiple-instance learning (MIL) framework, which reduces the gigapixel image analysis problem into detecting patches that contain the true signal while suppressing the noisy ones[110–114]. However, these methods can

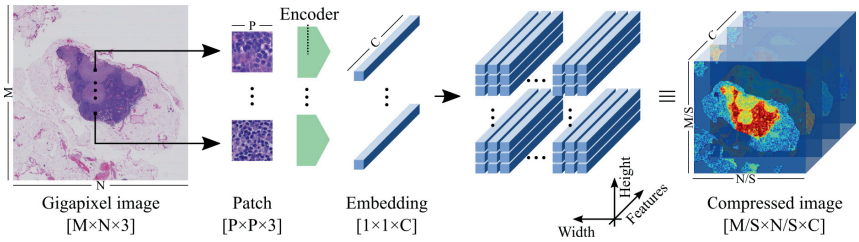

Figure 4.1: Gigapixel neural image compression. Left: a gigapixel histopathology whole-slide image is divided into a set of patches mapped to a set of low-dimensional embedding vectors using a neural network (the encoder). Center: these embeddings are stored keeping the spatial arrangement of the original patches. Right: the resulting array is a compressed representation of the gigapixel image. M and N: size of the gigapixel image; P: size of the square patches; C: size of the embedding vectors; and S: stride used to sample the patches. Typically: $M = N = 50{,}000$ and $P = S = C = 128$.

only take into account patterns present within individual patches, neglecting the potential relationships among them. More generally, MIL techniques cannot exploit patterns present in higher levels of abstraction since they ignore the spatial distribution among patches. This is also true for methods that aggregate patch-level information by means of spatial pooling[110,115].

In this work, we do not make any assumptions about the nature or spatial distribution of the visual cues associated with image-level labels. We argue that convolutional neural networks (CNN) are designed to solve the *what* and the *where* problems simultaneously[18], and propose a method to use them for gigapixel image analysis. However, feeding CNNs directly with gigapixel images is computationally unfeasible. Instead, we propose Neural Image Compression (NIC), a technique that maps images from a low-level pixel space to a higher-level latent space using neural networks. In this way, gigapixel images are compressed into a highly compact representation, which can be used to train a CNN using a single GPU for predicting any kind of image-level label.

4.1.2 Neural Image Compression

Gigapixel NIC was designed to reduce the size of a gigapixel image while retaining semantic information by shrinking its spatial dimensions and growing along the

feature direction (see Fig. 4.1). The method works by, first, dividing the gigapixel image into a set of high-resolution patches. Second, each high-resolution patch is compressed with a neural network (the *encoder*) that maps every image into a low-dimensional embedding vector. Finally, each embedding is placed into an array that keeps the original spatial arrangement intact so that neighbor embeddings in the array represent neighbor patches in the original image.

NIC was inspired by cognitive mechanisms. Human observers can describe complex visual patterns using only a few words without needing to describe each individual pixel. Similarly, the *encoder* can describe patches with low-dimensional embedding vectors, ignoring superfluous details. It is a powerful method that competes with classical approaches in terms of compression rate[116]. Moreover, previous works on representation learning and transfer learning have demonstrated that neural networks excel at extracting features that can be exploited by other networks to solve a variety of downstream tasks[117–120]. This makes NIC an ideal candidate for reducing the size of gigapixel images before feeding a CNN.

The *encoder* network can be trained using a wide variety of techniques. In this work, we selected and compared representative methods from three well-known families of unsupervised representation learning algorithms: reconstruction error minimization, contrastive training, and adversarial feature learning. First, autoencoders (AE) have been proposed as a straightforward method to learn a compact representation of a given data manifold[18]. AEs are neural networks that follow a particular encoder-bottleneck-decoder architecture. They aim to reconstruct input images by minimizing a reconstruction loss, e.g., the mean squared error (MSE). In particular, we considered the case of the variational autoencoder (VAE), a powerful modification of the original AE that relies on a probabilistic approach[87]. Second, we investigated a discriminative model based on contrastive training[119,121–123]. This model senses the world via an encoding network that maps images to embedding vectors. By training this model to distinguish between pairs of images with *same* or *different* semantic information, the encoder is enforced to learn a compact representation of the input data. Third, we investigated adversarial feature learning[117,118], a training framework based on Generative Adversarial Networks (GAN)[124]. GANs emerged as powerful generative models that map low-dimensional latent distributions into complex data. There is evidence that these latent spaces capture some of the high-level semantic information present in the data[125]. However, standard GAN models do not support the reverse operation, i.e., mapping data to the latent space. The Bidirectional GAN model (BiGAN[117]) learns this mapping using an explicit encoding

network in the training loop. Intuitively, the encoder benefits from all the high-level features which were fully automatically discovered by the generator.

4.1.3 Gigapixel Image Analysis

Without any loss of generality, we applied our method to two of the largest publicly available histopathology datasets to demonstrate its effectiveness in real-world applications: the *Camelyon16* Challenge[109] and the *TUPAC16* Challenge[82]. These datasets consist of gigapixel images of human tissue acquired with brightfield microscopy at very high magnification, also known as whole-slide images (WSI). These WSIs were stained with hematoxylin and eosin (H&E), the most widely used stain in routine histopathology diagnostics, that highlights general tissue morphology such as cell nuclei and cytoplasm. Each WSI is associated with a single image-level label: the presence of tumor metastasis for *Camelyon16*, and the tumor proliferation speed based on gene-expression profiling for *TUPAC16*.

A benefit of using a CNN for gigapixel image analysis is that, once trained, the CNN's areas of interest in the input image can be visualized using gradient-weighted class-activation maps (Grad-CAM)[126]. These saliency maps provide an answer to the *where* problem by locating visual cues related to the image-level labels. Identifying visual evidence for CNN predictions is of utmost importance in the medical domain regarding algorithm interpretation and knowledge discovery. For the first time, we performed this saliency analysis on gigapixel images and compared the resulting maps with the patch-level annotations of an expert observer.

4.1.4 Contributions

This work is an extension of our conference paper[127]. A number of additions have been made: three new datasets, an additional encoding method, the Grad-CAM analysis, a new experiment at the patch level, a new experiment at the image level, a more thorough evaluation using cross-validation, and an independent test evaluation performed by a third-party.

Our contributions can be summarized as follows:

- We propose Neural Image Compression (NIC) as a method to reduce gigapixel images to highly-compact representations, suitable for training a CNN end-to-

end to predict image-level labels using a single GPU and standard deep learning techniques.

- We compared several encoding methods that map high-resolution image patches to low-dimensional embedding vectors based on different unsupervised learning techniques: reconstruction error minimization, contrastive training, and adversarial feature learning.
- We evaluated NIC in three publicly available datasets: a synthetic set designed to evaluate the method; and two histopathological breast cancer sets of whole-slide images used to train the system to predict the presence of tumor metastasis and the tumor proliferation speed.
- We generated saliency maps representing the CNN's areas of interest in the image in order to discover and localize visual cues associated to the image-level labels.

The chapter is organized as follows: Sec. 4.2 and Sec. 4.3 describe the methods in depth; Materials and experimental results are described in Sec. 4.4; the discussions and conclusions are stated in Sec. 4.5 and Sec. 4.6, respectively.

4.2 Neural image compression

Let us define $\omega \in \mathbb{R}^{M \times N \times 3}$ as the gigapixel image (e.g., a WSI) to be compressed, with M rows, N columns, and three color channels (RGB). In order to compress ω into a more compact representation ω', two steps were taken. First, ω was divided into a set of high-resolution patches $X = \{x_{ij}\}$ with $x_{ij} \in \mathbb{R}^{P \times P \times 3}$, sampled from the i-th row and j-th column of an uniform grid of square patches of size P using a stride of S throughout ω. Second, each x_{ij} was compressed independently from each other, generating a set of low-dimensional embedding vectors of length C at each spatial location on the grid: $Y = \{e_{ij}\}$ with $e_{ij} \in \mathbb{R}^{C}$.

We formulated the task of mapping high-entropy X into low-entropy Y as an instance of an unsupervised representation learning problem, and parameterized this mapping function with a neural network E so that $X \xrightarrow{E} Y$. By sliding E throughout all ij spatial locations, ω was compressed into ω' with a total volume reduction of $F = 3\frac{S^2}{C}$. More formally:

$$\omega \in \mathbb{R}^{M \times N \times 3} \xrightarrow{E} \omega' \in \mathbb{R}^{\frac{M}{S} \times \frac{N}{S} \times C} \tag{4.1}$$

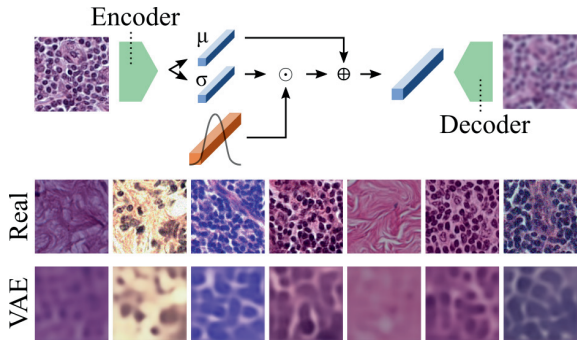

Figure 4.2: Variational Autoencoder. Top: the encoder maps a patch to an embedding vector depending on a noise vector while the decoder reconstructs the original patch from the embedding vector. Bottom: pairs of real and reconstructed patch samples using $C = 128$.

We investigated several unsupervised encoding strategies for learning E. Three of the most well-known and accessible methods in unsupervised image representation learning were selected. In all cases, neural networks were trained to solve an auxiliary task and learn E as a by-product of the training process. Note that none of the studied methods required the use of manual annotations. Network architectures and training protocols are detailed in the Supplementary Material at the end of this chapter.

4.2.1 Variational Autoencoder

Two networks are trained simultaneously, the encoder E and the decoder D. The task of E is to map an input patch x into a compact embedded representation e, and the task of D is to reconstruct x from e, producing x'. In this work, we used a more sophisticated version of AE, the variational autoencoder (VAE)[87]. The encoder in the VAE model learns to describe x with an entire probability distribution instead of a single vector (Fig. 4.2). More formally, E outputs $\mu \in \mathbb{R}^C$ and $\sigma \in \mathbb{R}^C$, two embeddings representing the mean and standard deviation of a normal distribution such that:

$$e = \mu + \sigma \odot n \tag{4.2}$$

with $n \sim \mathcal{N}(0, 1)$ and $\odot$ denoting element-wise multiplication.

We trained the VAE model by optimizing the following objective:

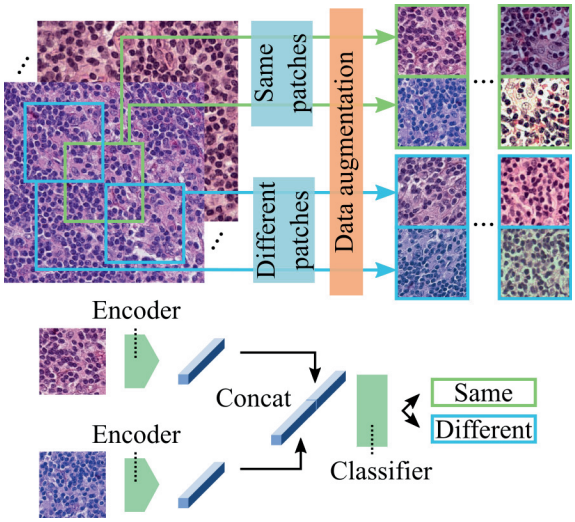

Figure 4.3: Contrastive training. Top: pairs of patches are extracted from gigapixel images. Pairs labeled as *same* originate from the same spatial location whereas *different* are extracted from either adjacent locations or different images. Bottom: scheme of a Siamese network trained for binary classification using the previous pairs.

$$\mathcal{V}_{\text{VAE}}(x, n, \theta_E, \theta_D) \quad = \quad \min_{E,D} \Big[\underbrace{\big(x - D(E(x,n))\big)^2}_{\text{Reconstruction error}} + \underbrace{\gamma(1 + \log\sigma^2 - \mu^2 - \sigma^2)}_{\text{KL divergence}} \Big] \quad (4.3)$$

with γ as a scaling factor, and θ_E and θ_D as the parameters of E and D, respectively. Note that we optimized θ_E and θ_D to minimize both the reconstruction error between the input and output data distributions, and the KL divergence between the embedding distribution and the normal $\mathcal{N}(0, 1)$ distribution.

This procedure results in a continuous latent space where changes in the embedding vectors are proportional to changes in the input data and vice-versa, effectively retaining semantic knowledge present in the input space.

4.2.2 Contrastive Training

We assembled a training dataset composed of pairs of patches $\boldsymbol{x} = \{x^{(1)}, x^{(2)}\}$ where each pair $\boldsymbol{x}$ was associated with a binary label y. Each label described whether the patches had been extracted from the *same* or a *different* location in a given gigapixel

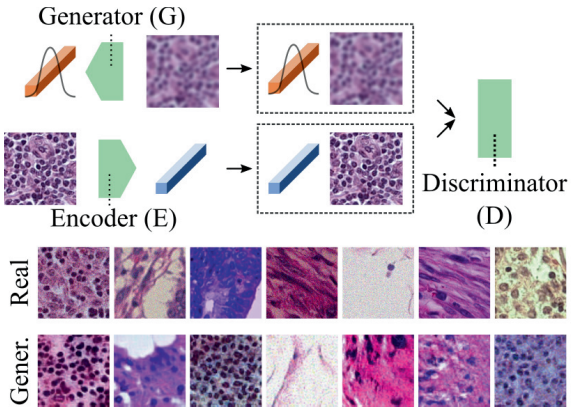

Figure 4.4: Adversarial Feature Learning. Top: three networks play a minimax game where the discriminator distinguishes between *actual* or *generated* image-embedding pairs, while the generator and the encoder fool the discriminator by producing increasingly more realistic images and embeddings. Bottom: real and generated patch samples using $C = 128$.

image, with $y = 1$ and $y = 0$, respectively. We trained a two-branch Siamese network[122] to solve this classification problem (Fig. 4.3).

We applied heavy data augmentation on all patches as indicated in[84], i.e., rotation, color augmentation, brightness, contrast, zooming, elastic deformation, and added Gaussian noise. Due to the strong augmentation, patches from the *same* location looked substantially different in a highly non-linear fashion while keeping a similar overall structure (semantic), see examples in Fig. 4.3. Patches from the *different* class were extracted from two distributions: 75% of them corresponded to non-overlapping adjacent locations (i.e., neighboring patches) where most of the visual features were shared, and the remaining 25% were sampled from different WSIs. Note that we included more of the neighboring *different* pairs to increase the difficulty of the classification task, forcing the network to extract higher-level features. The same data augmentation was applied to other encoders as well to ensure a fair comparison.

4.2.3 Bidirectional Generative Adversarial Network

The BiGAN setup consists of three networks: a generator G, a discriminator D, and an encoder E (Fig 4.4). G maps a latent variable $z \sim \mathcal{N}(0, 1)$ to generated images x':

$$z \sim \mathcal{N}(0,1) \in \mathbb{R}^C \xrightarrow{G} x' \in \mathbb{R}^{P \times P \times 3} \tag{4.4}$$

whereas E maps images x sampled from the true data distribution $\mathcal{X}$ to embeddings e:

$$x \sim \mathcal{X} \in \mathbb{R}^{P \times P \times 3} \xrightarrow{E} e \in \mathbb{R}^C \tag{4.5}$$

During training, the three networks play a minimax game where the discriminator D tries to distinguish between *actual* or *generated* image-embedding pairs, i.e., $\{x, e\}$ and $\{x', z\}$ respectively, while G and E try to fool D by producing increasingly more realistic images x' and embeddings e closer to $\mathcal{N}(0,1)$. More formally, we optimized the following objective function:

$$\begin{aligned}\mathcal{V}_{\text{BiGAN}}(x, z, \theta_G, \theta_E, \theta_D) = \\ = \min_{G,E} \max_{D} \Big[\log \big[D\big(x, \underbrace{E(x)}_{e}\big) \big] + \log \big[1 - D\big(\underbrace{G(z)}_{x'}, z\big) \big] \Big]\end{aligned} \tag{4.6}$$

with θ_G, θ_E, and θ_D representing the parameters of G, E, and D, respectively.

The authors of BiGAN theoretically and experimentally demonstrate that G and E learn an approximate inverse mapping function from each other, producing an encoding network E that learns a powerful low-dimensional representation of the image world inherited from G, suitable for downstream tasks such as supervised classification[117].

4.3 Gigapixel image analysis

In this section, we describe a method to train a CNN to predict image-level labels directly from compressed gigapixel images. Furthermore, we analyzed the location of visual cues associated with the image-level labels.

4.3.1 Feeding a CNN with compressed gigapixel images

We consider a dataset of gigapixel images $\Omega = \{\omega_i\}_{i=1}^{Q}$ that were compressed into $\Omega' = \{\omega'_i\}_{i=1}^{Q}$ with $\omega'_i \in \mathbb{R}^{\frac{M}{S} \times \frac{N}{S} \times C}$ using Eq. 4.1. In order to train a standard CNN on

a dataset like Ω', we set the depth of the convolutional filters of the input layer to be equal to the code size C used to compress the images.

We hypothesized that such a CNN can learn to detect highly discriminative features by exploiting two complementary sources of information from Ω': (1) the *global* context encoded within the spatial arrangement of embedding vectors, and (2) the *local* high-resolution information encoded within the features of each embedding vector.

4.3.2 Preventing overfitting

Note that in this setup, despite its gigapixel nature, each compressed image ω'_i constitutes a single training data point. Most public datasets with gigapixel images and their respective image-level labels consist only of a few hundred data points[82,109], increasing the risk of overfitting. The steps taken to prevent this effect are enumerated below.

First, we extended the training dataset Ω' by taking spatial crops of size $R \times R \times C$ from ω'_i, drastically increasing the total number and variability of the samples presented to the CNN[20]. During training, we randomly sampled the location of the center pixel of these crops. During testing, we selected T crops uniformly distributed along the spatial dimensions of ω'_i and averaged the predictions of the CNN across them[20]. Without any loss of generality, we applied this method to histopathology WSIs. As WSIs often contain large empty areas with no tissue, we detected the tissue regions[75] and sampled crops proportionally to the distance to background to accelerate the training, so that areas with higher tissue density were sampled more often. Similarly, test crops were sampled from locations where tissue was present.

The second measure taken to prevent overfitting was a simple augmentation at image level (i.e., 90-degree rotation and mirroring), encoding each image 8 times. This augmentation was carried out during testing as well, averaging the predictions of the CNN across them.

Finally, we designed a CNN architecture aimed at reducing the number of parameters present in the model. In particular, all convolutional layers were set to use depthwise separable convolutions, a type of convolution that reduces the number of parameters while maintaining a similar level of performance[46].

4.3.3 Visualizing visual cues related to image-level labels

The problem of feature localization is of utmost relevance for gigapixel image analysis: visual cues related to the image-level labels are often sparse and positioned in arbitrary locations within the image. For the purpose of identifying the location of these visual cues, we applied the Gradient-weighted Class-Activation Map (Grad-CAM) algorithm[126] to our trained CNN.

Given a compressed gigapixel image ω', its associated image-level label y, and a trained CNN, Grad-CAM performs a forward pass over ω' to produce a set of J intermediate three-dimensional feature volumes $f_j^{(k)}$, with j and k indicating the j-th and k-th convolutional layer and feature map, respectively. Subsequently, it computes the gradients of $f_j^{(k)}$ with respect to y for a fixed convolutional layer. It averages the gradients across the spatial dimensions and obtains a set of gradient coefficients $\gamma_j^{(k)}$, indicating how relevant each feature map is for the desired output y. Finally, it performs a weighted sum of the feature maps $f_j^{(k)}$ using the gradient coefficients $\gamma_j^{(k)}$:

$$h^{(k)} = \sum_{j=1}^{J} f_j^{(k)} \gamma_j^{(k)} \tag{4.7}$$

We applied the visualization method to the first convolutional layer ($k = 1$) in order to maximize the heatmap resolution.

4.4 Experimental results

We conducted a series of experiments to evaluate the performance of gigapixel NIC. First, we evaluated NIC in synthetic data to gain an understanding of the method and how its hyper-parameters affect performance. Second, we applied the method to several public histopathological datasets.

4.4.1 Materials

In this work, a synthetic dataset and three histopathology cohorts from different sources were used for supervised and unsupervised training at patch and image level; patients and WSIs were unique across all cohorts.

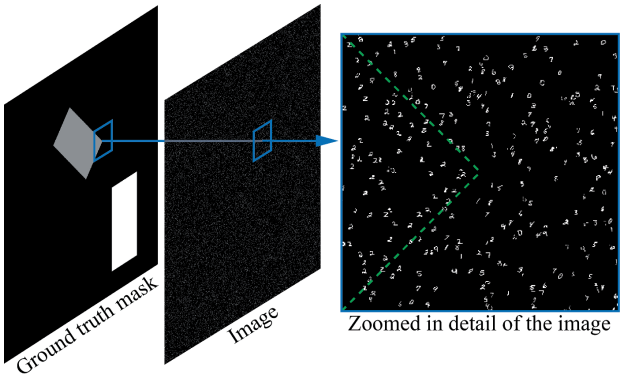

Figure 4.5: Example of an image from the synthetic dataset. Left: ground truth mask depicting the *tilted* and *non-tilted* rectangles that simulate lesions in grey and white, respectively. Center: image containing instances of MNIST digits; classes are defined by the rectangles or selected randomly. Right: all digits within the *tilted* rectangle boundary (in green) belong to the same class (number two), which corresponds to the image label as well.

Synthetic dataset

We developed and tested NIC with a synthetic dataset that mimicked the task of end-to-end WSI analysis before deploying it with real WSIs. As a substitute for WSIs, a set of images $T = \{t_i\}$ with $t_i \in \mathbb{R}^{A \times B}$ were used, each one associated with a dense pixel-level ground truth mask $M = \{m_i\}$, where $m_i \in \mathbb{R}^{A \times B}$, and an image-level scalar label $Y = \{y_i\}$.

To emulate global patterns in the images (e.g., tumor lesions), we defined two rectangles within each mask placed at random locations and characterized by their own orientation: one was either vertically or horizontally oriented (*non-tilted*); the other was tilted either 45 or 135 degrees (*tilted*). Each rectangle was associated to a randomly selected MNIST[128] digit class. To emulate local patterns (e.g., cells), instances of MNIST digits were placed throughout the images at random locations. The class of these instances was determined by their spatial position, i.e., belonging to a certain rectangle class if placed within the boundaries of a rectangle or otherwise randomly selected. The label of each image was defined by the class of the *tilted* rectangle, with the *non-tilted* rectangle acting as a distraction. See Fig. 4.5 for an example image.

Note that, in order to solve this classification task, NIC had to detect the *tilted* rectangle and its class without access to the ground-truth masks. Moreover, the method

must combine local and global information, i.e., exploiting the local features that identify digit instances' classes while recognizing their global spatial arrangement to detect the orientation of the rectangle.

We downsampled MNIST digits to 9×9 pixels, defining a patch size $P = 9$ and stride $S = 9$ pixels. WSIs are typically 50000×50000 pixels in size, with patch sizes of 128×128 pixels covering structures composed of a few cells. We mimicked this image-patch ratio by using an image size of $A = B = 3600$ pixels, and inserted $25{,}920$ digit instances per image (0.2% of the total possible locations). Rectangle size randomly ranged from 1800 pixels to 36 pixels (long side). This reduced image size enabled us to run more thorough experiments than what we could do with histopathological data.

A total of $50{,}000$ images with balanced labels were created across the 10 digit classes: 2500 to generate patches to train the encoders, $22{,}500$ to train the NIC CNN (with 75% and 25% for training and validation), and $25{,}000$ as an independent test set for the NIC CNN.

Camelyon16 histopathology dataset

The *Camelyon16*[109] dataset is a publicly available multicenter cohort that consists of 400 sentinel lymph node H&E WSIs from breast cancer patients. Reference standard exists in two forms: fine-grained annotations of metastatic lesions and image-level labels indicating the presence of tumor metastasis in each slide. Sixty WSIs from the original training set were set aside to train encoders at patch level. The remaining WSIs were combined with the original test set (n=340) to train and evaluate a classification model using image-level labels only.

TUPAC16 histopathology dataset

The *TUPAC16*[82] dataset was used, consisting of 492 H&E WSIs from invasive breast cancer patients. It is a publicly available cohort with WSIs from The Cancer Genome Atlas[73] where each WSI is associated with a tumor proliferation speed score, an objective measurement that takes into account the RNA expression of 11 proliferation-associated genes[74]. We set aside 40 WSIs from this set to train encoders at patch level. The remaining WSIs (n=452) were used to train and evaluate a regression model using image-level labels only. Additionally, 321 test WSIs with no public

ground truth available were used to perform an independent evaluation.

Rectum histopathology dataset

The *Rectum* dataset is a publicly available set of 74 H&E WSIs from rectal carcinoma patients[98]. Manual annotations of 9 tissue classes were made by an expert: blood cells, fatty tissue, epithelium, lymphocytes, mucus, muscle, necrosis, stroma, and tumor. The slides were randomized and organized into ten equal partitions at patient level, five of which were used for training, one for validation, and four for testing. This dataset was used to train and evaluate encoders at patch level only. We extracted a balanced distribution of 15K, 852, and 4K patches per class from the training, validation, and test slides, respectively.

Data preparation

Regarding the synthetic dataset, one million pairs of patches were extracted to train the encoders, augmented with scaling and elastic deformation. To avoid creating a dataset of empty patches, the probability of sampling a patch containing a white pixel was twice of that of an empty patch.

All WSIs in this study were preprocessed with a tissue-background segmentation algorithm[75] in order to exclude areas not containing tissue from the analysis. Furthermore, all images were analyzed at $0.5\,\mu\mathrm{m/pixel}$ resolution.

A set of patch datasets were assembled to train and evaluate each of the encoding networks described in Sec. 4.2 using the set of images that we set aside from each cohort: 60 WSIs from *Camelyon16*, 40 from *TUPAC16*, and all from *Rectum*. Each of these subcohorts were divided into training, validation, and test partitions.

The *contrastive dataset* was created by extracting an equal amount of patches from each source (i.e., *Camelyon16*, *TUPAC16*, and *Rectum*) and merged into $50{,}000$ and $25{,}000$ patch pairs for training and validation, respectively. The *non-contrastive dataset* was then created by randomizing all individual patches within the *contrastive dataset*.

The *supervised-tumor dataset* was created by extracting $50{,}000$, $10{,}000$, and $50{,}000$ patches from the set of 60 *Camelyon16* WSIs for training, validation, and testing, respectively. Finally, the *supervised-tissue dataset* consisted of the *Rectum* training, validation, and

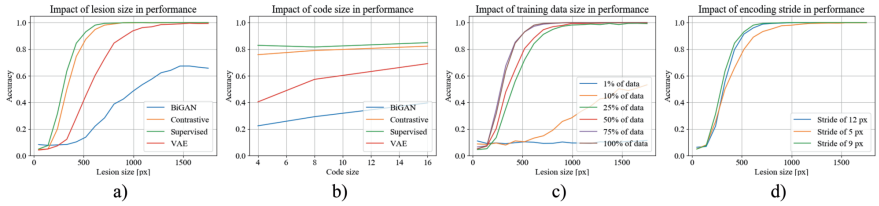

Figure 4.6: Experimental results with synthetic data and image-level labels. Default hyper-parameter choice unless specified otherwise is: *supervised* encoder, code size 16, stride 9 pixels, and usage of 100% of training data.

test sets containing $131{,}000$, 8000, and $35{,}000$ patches, respectively. Note that the patches in the *supervised-tumor dataset* and *supervised-tissue dataset* test sets did not undergo any augmentation. The fine-grained tumor annotations were used to sample a balanced distribution of tumor and non-tumor patches in the *supervised-tumor dataset* and 9-class patches in the *supervised-tissue dataset*.

4

4.4.2 Experimental results on synthetic data

The *contrastive* encoder was trained using the pairs of patches described in Sec. 4.4.1. The *VAE* and *BiGAN* encoders were subsequently trained using these same patches, concatenating and shuffling them along the pair dimension. Finally, a *supervised* encoder was trained with MNIST digits to serve as an oracle feature extractor. Once the encoders were trained, all images were encoded to produce a different embedded representation for each encoding configuration. Network architectures and training protocols are detailed in the Supplementary Material at the end of this chapter.

We explored different values for the method hyper-parameters (e.g., code size and stride) using the synthetic data, and evaluated the accuracy of each resulting CNN in the independent test set. We analyzed how this performance was affected by the size of the simulated lesion, i.e., the size of the *tilted* rectangle. Results are summarized in Fig. 4.6. Overall, the *contrastive* encoder achieved the best performance among the unsupervised techniques, very close to that of the oracle, followed by the *VAE* and *BiGAN* encoders. This trend was maintained when analyzing the impact of the lesion size. We found out that the method's performance degraded quickly when the size of the target lesion was smaller than 10% of the image size (see Fig. 4.6-a).

Additionally, the performance impact of the code size used to compress the images

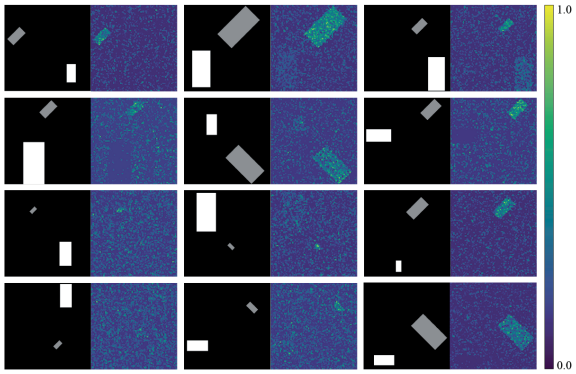

Figure 4.7: Grad-CAM visualization applied to randomly selected synthetic test images. Left images within the pairs correspond to the ground truth masks (unseen by the model), and right ones to the saliency heatmaps. Note that areas corresponding to the grey *tilted* rectangles (responsible for the image-level labels) are highly salient with respect to the rest of the image.

was assessed (Fig. 4.6-b). It was observed that larger code sizes generally improved performance, a result that was more evident for less accurate encoding methods like *VAE* and *BiGAN*. Subsequently, different stride values were tested using the oracle encoder and a code size of 16: it was found that a smaller stride, producing embedded images with larger spatial resolution, resulted in hampered performance (Fig. 4.6-c). Finally, the impact of training data size in performance was analyzed using the oracle encoder with code size 16 and stride 9 (Fig. 4.6-d). These results indicate that NIC required in the order of thousands of images to perform well, a requisite that is rarely met in real histopathological datasets.

In our last experiment, we applied Grad-CAM to visualize the regions of the input images that were responsible for the CNN prediction (see Fig. 4.7). Remarkably, the network seemed to be able to discern between background noise and the rectangular patterns. Upon visual inspection, the CNN generally focused on the *tilted* rectangle, the one responsible for the image-level label. We applied a simple general-purpose post-processing routine to denoise the heatmaps and reject spurious activity. We measured the Jaccard similarity coefficient per image between the post-processed heatmap and the ground truth maps, and obtained 0.612 on average across test images.

4.4.3 Training of encoders

Due to the computationally expensive nature of experimenting with gigapixel WSIs, we only tested a subset of the hyper-parameters that we explored with synthetic data. We selected their values using the following heuristics. We used $P = 128$, a common patch size used in the Computational Pathology literature[75], with a stride of the same size $S = 128$ to perform non-overlapping patch sampling. We selected $R = 400$ to obtain crops corresponding to typical sizes of gigapixel WSIs ($50{,}000 \times 50{,}000$ pixels) and $T = 10$ as done in the literature[20]. Finally, we selected $C = 128$ to perform our experiments using a single GPU. Network architectures and training protocols are detailed in the Supplementary Material.

We trained the *contrastive* encoder with the *contrastive dataset*, and the *VAE* and *BiGAN* models with the *non-contrastive dataset*. Note that these datasets contained the exact same image patches, ensuring a fair comparison among encoders. No manual annotations were required in this process. We trained a supervised baseline encoder for breast tumor classification using the *supervised-tumor dataset*, and a supervised baseline encoder for rectum tissue classification using the *supervised-tissue dataset*.

It is widely recognized that color-based features can be very informative in histopathology image analysis[129–131]. Therefore, we included an additional encoding function to capture color information from the raw input by averaging the pixel intensity across spatial dimensions from input RGB patches. It provided a simple yet effective baseline to compare with more sophisticated encoding mechanisms.

This entire training process resulted in 6 encoding networks used in subsequent experiments: the *mean-RGB* baseline, *VAE* encoder, *contrastive* encoder, *BiGAN* encoder, *supervised-tumor* baseline, and *supervised-tissue* baseline.

4.4.4 Comparing encoding performance

Due to the lack of a common evaluation methodology for unsupervised representation learning, we compared the performance of these 6 encoders when used as fixed feature extractors for related supervised classification tasks. We defined two tasks: (1) discerning between tumor and non-tumor patches on the *supervised-tumor dataset* (*Task-1*), and (2) performing 9-class tissue classification on the *supervised-tissue dataset* (*Task-2*). For each task, we trained an MLP on top of each encoder with frozen weights and reported the accuracy metric for each test set.

Results in Tab. 4.1 highlight several observations. First, *VAE*, *contrastive*, and *BiGAN* performed better than the lower baseline for both *Task 1* and *Task 2*, stressing their ability to describe complex patterns beyond simple features related to color intensity. Second, the *VAE* encoder obtained a higher performance than the *contrastive* one, particularly for *Task 2*. Third, the *BiGAN* encoder achieved the best performance among all the unsupervised methods, with a relatively large margin for the more complex *Task 2* with respect to the runner-up *VAE* model. Furthermore, the *BiGAN* encoder obtained the best result for 5 out of 9 classes in *Task 2*, and it achieved the first or second best result for 8 out of 9 classes among the unsupervised models. Remarkably, *BiGAN* succeeded at classifying patches from challenging tissue classes such as blood cells and necrotic tissue.

4.4.5 Predicting the presence of metastasis at image level

In this experiment, we trained a CNN to perform binary classification on compressed gigapixel WSIs from the *Camelyon16* cohort, identifying the presence of tumor metastasis using image-level labels only. Due to the limited amount of images in this cohort (340 WSIs), we divided the dataset into four equal-sized partitions and performed four rounds of cross-validation using two partitions for training, one for validation and one for testing, rotating them in each round. We trained a different CNN classifier for each encoder, i.e., *mean-RGB*, *VAE*, *contrastive*, *BiGAN*, and the upper baseline *supervised-tumor*. We reported the area under the receiver operating characteristic (AUC) on three evaluation sets.

The first evaluation set (*All*) concatenated all samples in each of the hold-out partitions. Note that each hold-out partition was evaluated by a different CNN that had never seen the data. The second evaluation set (*Test*) was a subset of *All* that matched the official test set of the *Camelyon16* Challenge, used for comparison with the public leaderboard. The third evaluation set (*Macro*) used the same WSIs as in *Test* but considering only those that presented a macro metastasis as positive labels, i.e., a tumor lesion larger than $2\,\mathrm{mm}$. The macro labels were only available for the *Camelyon16* test set. The *Macro* set was relevant to evaluate how the method performed with lesions visible at low resolution.

Results in Tab. 4.2 demonstrate that the method presented in this work is an effective technique for gigapixel image analysis using image-level labels only. Regarding

Table 4.1: Patch-level classification performance (accuracy). *Task-1* and *Task-2* in the text refer to columns *Camelyon-Tumor* and *Rectum-Global*. Reporting mean and standard deviation using two random weight initializations.

	Camelyon					**Rectum**					
Encoder	**Tumor**	**Blood**	**Fat**	**Epith**	**Lymph**	**Mucus**	**Muscle**	**Necro**	**Strom**	**Tumor**	**Global**
VAE	0.799(0.004)	0.602(0.034)	0.735(0.154)	0.556(0.006)	0.811(0.018)	0.623(0.125)	**0.823(0.014)**	0.170(0.018)	**0.768(0.008)**	0.667(0.000)	0.639(0.010)
Contrastive	0.789(0.004)	0.304(0.018)	**0.966(0.003)**	0.502(0.005)	0.850(0.014)	0.240(0.011)	0.609(0.006)	0.140(0.010)	0.595(0.005)	0.476(0.014)	0.520(0.002)
BiGAN	**0.806(0.022)**	**0.738(0.034)**	0.879(0.059)	**0.627(0.000)**	**0.899(0.008)**	**0.802(0.055)**	0.796(0.002)	**0.769(0.021)**	0.601(0.066)	**0.770(0.010)**	**0.765(0.013)**
Mean-RGB	0.772(0.001)	0.736(0.000)	0.635(0.355)	0.202(0.049)	0.385(0.068)	0.720(0.270)	0.904(0.008)	0.030(0.028)	0.668(0.039)	0.252(0.000)	0.504(0.022)
Sup.-tumor	0.855(0.001)	0.578(0.090)	0.896(0.005)	0.400(0.007)	0.981(0.004)	0.868(0.021)	0.507(0.061)	0.494(0.049)	0.467(0.027)	0.618(0.019)	0.646(0.008)
Sup.-tissue	0.800(0.006)	0.835(0.003)	0.958(0.008)	0.832(0.029)	0.935(0.010)	0.937(0.026)	0.940(0.002)	0.906(0.005)	0.863(0.009)	0.934(0.002)	0.904(0.000)

Table 4.2: Predicting the presence of metastasis at WSI level (AUC). Reporting mean and standard deviation using two random weight initializations.

Encoder	All	Test	Macro
VAE	0.661(0.007)	0.671(0.008)	0.634(0.003)
Contrastive	0.608(0.001)	0.651(0.016)	0.606(0.012)
BiGAN	**0.725(0.009)**	**0.704(0.030)**	**0.720(0.010)**
Mean-RGB	0.582(0.006)	0.578(0.016)	0.585(0.014)
Supervised-tumor	0.760(0.002)	0.771(0.002)	0.914(0.000)

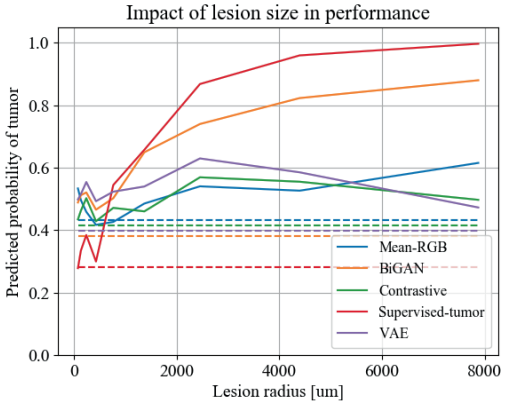

Figure 4.8: Experimental results with respect to lesion size in *Camelyon16 all* test set using multiple encoders. Solid lines: average probability of samples with positive labels; dashed lines: average probability of samples with negative labels (no lesion).

the *All* evaluation set, *BiGAN* achieved a remarkable performance of 0.716 AUC, with a relative difference from the *supervised* baseline of only 6%. The *contrastive* and *VAE* models also surpassed the lower baseline, but obtained substantially lower performance scores compared to *BiGAN*. Regarding the *Test* set, the *BiGAN* encoder obtained a lower performance of 0.674 AUC. In the *Macro* set, the performance gap between the *supervised* baseline and the *BiGAN* encoder increased substantially from 0.095 to 0.184. The state-of-the-art in *Camelyon16* obtained 0.9935 AUC in the *Test* set using accurate pixel-level annotations to train their model.

Additionally, we analyzed the performance of our method as a function of the lesion size in the *All* test set. The lesion size is a measurement determined by pathologists taking the distribution of tumor cell clusters within a WSI into account. Since this annotation was not available for all WSIs, we approximated it by computing

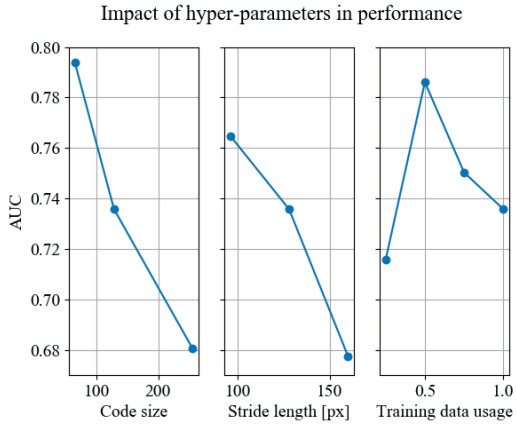

Figure 4.9: Hyper-parameter value analysis performed in *Camelyon16* data using the *supervised* encoder. Evaluated on unseen images from the first data partition out of the 4-fold cross-validation sets. Left: varying code size using a fix stride of 128 pixels; center: varying stride while using a fix code size of 128 elements; and right: varying the number of WSIs used during training.

the radius of an hypothetical circle with an area composed of all pixels annotated as tumor in each WSI. Results in Fig. 4.8 indicated that our method's performance degraded with small tumor lesions across most encoders, in line with the results obtained with synthetic data. Furthermore, we experimented with different hyper-parameters such as code size, stride, and training data size using the *supervised* encoder (Fig. 4.9). We found that performance improvements might be gained from careful hyper-parameter tuning of the code size and stride parameters. Moreover, there seemed to be a weak but positive correlation between model performance and training data size.

4.4.6 Predicting tumor proliferation speed at image level

In this experiment, we trained a CNN to perform a regression task on compressed gigapixel WSIs from the *TUPAC16* cohort, predicting the tumor proliferation speed based on gene-expression profiling. We performed 4-fold cross-validation as in the previous experiment, and reported the Spearman correlation between the predicted and the true scores of two evaluation sets.

The first evaluation set (*All*) concatenated all samples in each of the hold-out partitions. The second evaluation set (*Test*) matched the test set used in the *TUPAC16*

Table 4.3: Predicting tumor proliferation speed at WSI level (Spearman corr.). Reporting mean and standard deviation using two random weight initializations.

Encoder	All	Test
VAE	0.419(0.004)	-
Contrastive	0.390(0.006)	-
BiGAN	**0.522(0.001)**	0.558(0.001)
Mean-RGB	0.238(0.020)	-
Supervised-tumor	0.427(0.014)	-

Challenge, whose ground truth is not public. Using the encoder that obtained the highest performance, we evaluated each WSI in *Test* four times using each of the CNNs trained during cross-validation and submitted the average score per slide. Our predictions were independently evaluated by the challenge organizers, ensuring a fair and independent comparison with the state of the art.

The results in Tab. 4.3 showed that *BiGAN* achieved the highest performance with a 0.521 Spearman correlation. Remarkably, this score was superior to that of any other unsupervised or supervised encoder. In addition, we obtained a score of 0.557 on the *TUPAC16* Challenge test set, superior to the state-of-the-art for image-level regression with a score of 0.516. Note that the first entry of the leaderboard used an additional set of manual annotations of mitotic figures, thus it cannot be compared with our setup.

4.4.7 Visualizing *where* the information is located

We conducted a qualitative analysis on the trained CNNs to locate the spatial position of visual cues relevant in predicting the image-level labels. We applied the Grad-CAM algorithm to the CNNs trained for both tasks at image level. For the tumor metastasis prediction task, we compared the saliency maps with fine-grained manual annotations. Figures 4.10 and 4.11 include the results for a few samples; the results for the remaining WSIs can be found in the Supplementary Material. Note that each WSI was evaluated by a CNN that had not yet seen the image (hold-out partition).

Fig. 4.10 shows that the *mean-RGB* baseline model lacked the ability to focus on specific tissue regions, suggesting that it was unable to learn discriminative features

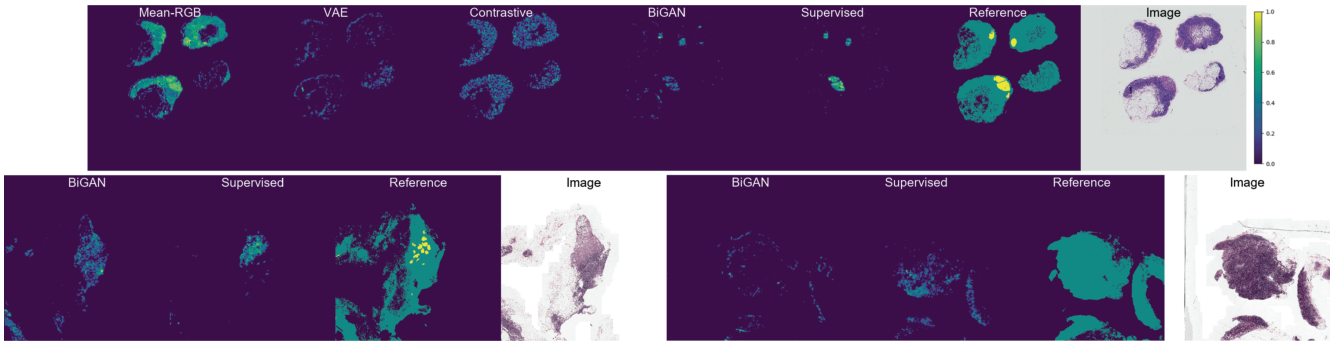

Figure 4.10: Grad-CAM visualization applied to several WSIs from *Camelyon16*. Top: the first five images represent the saliency maps for CNNs trained with 5 different encoders, respectively. The sixth and seventh images are the reference standard (manual annotations) and RGB thumbnail of the WSI, respectively. Dark blue represents low saliency, whereas yellow indicates high saliency. Bottom-left: failure case where the *BiGAN* model failed to recognize the tumor area. Bottom-right: failure case where the *BiGAN* model attended to a region with no tumor cells.

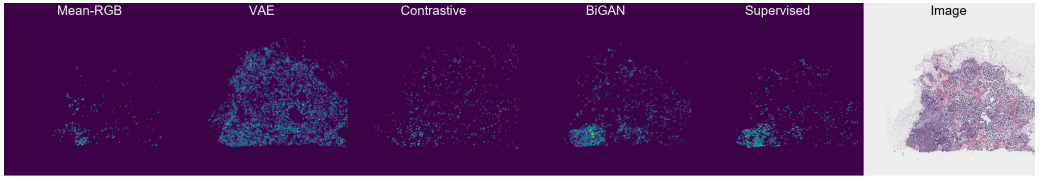

Figure 4.11: Grad-CAM visualization applied to a sample case from *TUPAC16*. The first five images represent the saliency maps for CNNs trained with five different encoders, respectively. The last image is an RGB thumbnail of the WSI. Dark blue represents low saliency, whereas yellow indicates high saliency.

from image-level labels. The *VAE* and *contrastive* models exhibited a suboptimal behavior, scattering attention all over the image. Remarkably, the *BiGAN* model seemed to focus on tumor regions only, discarding empty areas, fatty tissue, and healthy dense tissue. It showed a strong discriminative power to discern between tumor and non-tumor regions, even though the CNN had access to image-level labels only. For completeness, we also included the *supervised-tumor* baseline that also exhibited a focus on tumor regions. Nevertheless, these heatmaps are often difficult to interpret and cannot be used for a more quantitative analysis. Failure cases can be seen in the bottom part of Fig. 4.10, where the CNN highlighted non-tumorous regions.

Regarding Fig. 4.11, a similar trend to the one found in the previous task was observed for all encoders: the *BiGAN* model focused on very specific regions of the WSIs that seemed compatible with active tumor regions. The *supervised-tumor* baseline focused on irrelevant areas, in line with its poor performance for this task.

4.5 Discussion

Our experimental results support the hypothesis that visual cues associated with weak image-level labels can be exploited by our method, integrating information from global structure and local high-resolution visual cues. Furthermore, we have shown that this methodology is flexible and completely label-agnostic, delivering relevant results for both classification and regression tasks in synthetic as well as histopathological data. It emerges as a promising strategy to tackle the analysis of more challenging image-level labels that are closely related to patient outcome, e.g., overall survival and recurrence-free survival. Gigapixel NIC paves the way for leveraging existing computer vision algorithms that could not be applied in the gigapixel domain until now, such as image captioning (useful to generate written clinical reports), visual question answering, image retrieval (to find similar pathologies), anomaly detection, and generative modeling[132–136].

A key assumption in our method was that high-resolution image patches could be represented by low-dimensional highly compressed embedding vectors. We analyzed several unsupervised strategies to achieve such a compression and found that the *BiGAN* encoder, trained using adversarial feature learning, was superior to all other methods across all experiments with histopathological data. We believe that this relative improvement with respect to the *VAE* and *contrastive* methods is

explained by intrinsic algorithmic differences among the methods. In particular, the *VAE* model relies on minimizing the MSE objective, which is a unimodal function that fails to capture high-level semantics; it focuses on reconstructing low-level pixel information instead, wasting embedding capacity. On the other hand, the *contrastive* encoder uses the embedding capacity more efficiently, but its performance is driven by the design of the hand-engineered contrastive task. Remarkably, the *BiGAN* model learns an encoder that fully automatically inverts a complex mapping between the latent space and the image space. By doing so, the encoder benefits from all the high-level features and semantics already discovered by the generator, producing very effective discriminative embedding vectors. Furthermore, *BiGAN* achieved the best classification accuracy on the challenging blood, mucus, and necrotic tissue classes that rarely appear in the *Camelyon16* and *TUPAC16* WSIs. We hypothesize that the adversarial method can model these rare data modes more effectively than the *contrastive* or *VAE* approaches. Nevertheless, we believe that the choice of encoder may be data-dependent, since the *contrastive* encoder outperformed the other approaches in the synthetic dataset.

We trained a CNN to predict the breast tumor proliferation speed based on gene-expression profiling, a label associated with unknown visual cues. Our method succeeded in finding and exploiting these patterns in order to predict expected tumor proliferation speed, surpassing the current state-of-the-art for image-level based methods. This shows that our method constitutes an effective solution to deal with gigapixel image-level labels with unknown associated visual cues. Moreover, our method could be used in future works to effectively mine datasets with thousands of gigapixel images[115]; other automatically generated labels from immunohistochemistry, genomics, or proteomics can be targeted, and visual patterns beyond the knowledge of human pathologists may be discovered.

For the first time, the regions of a gigapixel image that a trained CNN attends to when predicting image-level labels were visualized, and the effect of different encoding methods was compared. We discovered that only the CNNs trained with images compressed with the *BiGAN* encoder and the *supervised-tumor* baseline were able to attend to regions of the image where tumor cells were present. The fact that the *BiGAN* model simultaneously learned to delimit metastatic lesions and identify tumor features within the patch embeddings validates our hypothesis that CNNs are an effective method for analyzing gigapixel images, i.e., since they can exploit both global and local context.

We targeted the presence of tumor metastasis in breast lymph nodes and showed that the *BiGAN* setup performed similarly to the *supervised* baseline. However, our best-performing algorithm was still inferior to that of the *Camelyon16* leadingboard (0.9935 AUC using accurate pixel-level annotations). This performance gap is likely due to two factors. First, the majority of the images marked as positive contain tumor lesions comprised of only a few tumor cells (i.e., micro-metastasis), becoming almost undetectable with the compression setup tested in this work (see Fig. 4.8). Second, the lack of training data (only a few hundred training images) may lead the CNN into the overfitting regime.

We acknowledge several limitations of our method. For one, it requires a substantial amount of I/O throughput and storage due to the need to write compressed WSI representations to disk before training, and repetitively read them to assemble mini-batches during training. This computational burden prevented us from performing a wide hyper-parameter value search, which may have resulted in a suboptimal parameter selection. Second, it was also observed that the method's performance was proportional to lesion size. In particular, it struggled to detect micro-metastasis in *Camelyon16* data, i.e., tumor lesions smaller than $2\,\mathrm{mm}$, limiting the applicability of NIC to tasks with large lesions.

This method can be extended in multiple ways. More sophisticated encoders may improve the low-dimensional representation of the image patches[119,137,138]. Incorporating attention mechanisms may make it easier for the CNN to attend to relevant regions for the image-level labels[139], improving the detection of small lesions. Finally, gradient checkpointing[140] could be used to backpropagate the training signal from the image-level labels towards the encoder weights.

4.6 Conclusion

Our method for gigapixel neural image compression was able to distill relevant information into compact image representations. The fact that a CNN could be trained using these alternative learned representations opens opportunities to use other methods: gigapixel images are no longer considered as low-level pixel arrays, but operate in a higher level of abstraction. In this work, we showed examples of classification, regression, and visualization performed in a latent space learned by a neural network. These positive results enable performing more advanced gigapixel applications in the latent space, such as data augmentation, generative modeling,

content retrieval, anomaly detection, and image captioning.

4.7 Acknowledgment

This study was supported by a Junior Researcher grant from the Radboud Institute of Health Sciences (RIHS), Nijmegen, The Netherlands; a grant from the Dutch Cancer Society (KUN 2015-7970); and another grant from the Dutch Cancer Society and the Alpe d'HuZes fund (KUN 2014-7032); this project has also been partially funded by the European Union's Horizon 2020 research and innovation programme under grant agreement No 825292.

The authors would like to thank Dr. Mitko Veta for evaluating our predictions in the test set of the *TUPAC16* dataset, and the developers of Keras[141], the open source tool that we used to run our deep learning experiments.

4.8 Appendix

4.8.1 Encoders for Histopathological Data

Variational Autoencoder

Two networks are trained simultaneously, the encoder E and the decoder D. The task of E is to map an input patch $x \in \mathbb{R}^{P\times P\times 3}$ to a compact embedded representation $e \in \mathbb{R}^{C}$, and the task of D is to reconstruct x from e, producing $x' \in \mathbb{R}^{P\times P\times 3}$. In this work, we used a more sophisticated version of AE, the variational autoencoder (VAE)[87], with $P = 128$ and $C = 128$.

The encoder in the VAE model learns to describe x with an entire probability distribution, in particular, given an input x, the encoder E outputs $\mu \in \mathbb{R}^{C}$ and $\sigma \in \mathbb{R}^{C}$, two embeddings representing the mean and standard deviation of a normal distribution so that:

$$e = \mu + \sigma \odot n \tag{4.8}$$

with $n \sim \mathcal{N}(0, 1)$ and $\odot$ denoting element-wise multiplication.

The architecture of E consisted of 5 layers of strided convolutions with 32, 64, 128, 256 and 512 3×3 filters, batch normalization (BN) and leaky-ReLU activation (LRA); followed by a dense layer with 512 units, BN and LRA; and a linear dense layer with C units.

The architecture of the decoder D consisted of a dense layer with 8192 units, BN and LRA, eventually reshaped to $(4 \times 4 \times 512)$; followed by 5 upsampling layers, each composed of a pair of nearest-neighbor upsampling and a convolutional operation[106], with 256, 128, 64, 32 and 16 3×3 filters, BN and LRA; finalized with a convolutional layer with 3 3×3 filters and tanh activation.

We trained the VAE model by optimizing the following objective:

$$\mathcal{V}_{\text{VAE}}(x, n, \theta_E, \theta_D) \quad = \quad \min_{E,D} \Big[\underbrace{\big(x - D(E(x,n))\big)^2}_{\text{Reconstruction error}} + \underbrace{\gamma(1 + \log\sigma^2 - \mu^2 - \sigma^2)}_{\text{KL divergence}} \Big] \quad (4.9)$$

with x representing a single data sample, n a sample from $\mathcal{N}(0,1)$, γ a scaling factor, and θ_E and θ_D as the parameters of E and D, respectively. Note that we optimized θ_E and θ_D to minimize both the reconstruction error between the input and output data distributions, and the KL divergence between the embedding distribution and the normal $\mathcal{N}(0,1)$ distribution with $\gamma = 5 \times 10^{-5}$.

We minimized $\mathcal{V}_{\text{VAE}}$ using stochastic gradient descent with Adam optimization and 64-sample mini-batch, decreasing the learning rate by a factor of 10 starting from 1×10^{-3} every time the validation loss plateaued until 1×10^{-5}. Finally, we selected the encoder E corresponding to the VAE model with the lowest validation loss.

Contrastive Training

We assembled a training dataset composed of pairs of patches $\boldsymbol{x} = \{x^{(1)}, x^{(2)}\}$ with $x^{(i)} \in \mathbb{R}^{P \times P \times 3}$ where each pair $\boldsymbol{x}$ was associated with a binary label y, and $P = 128$. In order to solve this binary classification task, we trained a two-branch Siamese network[122] called S. Both input branches shared weights and consisted of the same encoding architecture E as the VAE model. After concatenation of the resulting embedding vectors, a MLP followed consisting of a dense layer with 256 units, BN and LRA; finalized by a single sigmoid unit.

We minimized the binary cross-entropy loss using stochastic gradient descent with Adam optimization and 64-sample mini-batch, decreasing the learning rate by a factor of 10 starting from 1×10^{-2} every time the validation classification accuracy plateaued until 1×10^{-5}. Finally, we selected the encoder E corresponding to the S with the highest validation classification accuracy.

Bidirectional Generative Adversarial Network

We trained a BiGAN setup consisting of three networks: a generator G, a discriminator D and an encoder E. G mapped a latent variable z drawn from a normal distribution $\mathcal{N}(0,1)$ into artificial images x':

$$z \sim \mathcal{N}(0,1) \in \mathbb{R}^C \xrightarrow{G} x' \in \mathbb{R}^{P \times P \times 3} \tag{4.10}$$

whereas E mapped images x sampled from the true data distribution $\mathcal{X}$ into embeddings e:

$$x \sim \mathcal{X} \in \mathbb{R}^{P \times P \times 3} \xrightarrow{E} e \in \mathbb{R}^C \tag{4.11}$$

During training, the three networks played a minimax game where the discriminator D tried to distinguish between *actual* and *artificial* image-embedding pairs, i.e. $\{x, e\}$ and $\{x', z\}$ respectively, while G and E tried to fool D by producing increasingly more realistic images x' and embeddings e, i.e. closer to $\mathcal{N}(0,1)$. We used $P = 128$ and $C = 128$.

Given the difficulty of training a stable BiGAN model, we downsampled x by a factor of 2 before feeding it to the model. The architecture of the encoder E consisted of 4 layers of strided convolutions with 128 3×3 filters, BN and LRA; followed by a linear dense layer with C units.

The architecture of the generator G consisted of a dense layer with 1024 units, BN and LRA, eventually reshaped to $(4 \times 4 \times 64)$; followed by 4 upsampling layers, each composed of a pair of nearest-neighbor upsampling and a convolutional operation[106], with 128 3×3 filters, BN and LRA; finalized with a convolutional layer with 3 3×3 filters and tanh activation.

The discriminator D had two inputs, a low-dimensional vector and an image. The image was fed through a network with an architecture equal to E but different

weights, and the resulting embedding vector concatenated to the input latent variable. This concatenation layer was followed by two dense layers with 1024 units, LRA and dropout (0.5 factor); finalized with a sigmoid unit.

We optimized the following objective function:

$$\mathcal{V}_{\text{BiGAN}}(x, z, \theta_G, \theta_E, \theta_D) = \\ = \min_{G,E} \max_{D} \Big[\log \big[D\big(x, \underbrace{E(x)}_{e}\big) \big] + \log \big[1 - D\big(\underbrace{G(z)}_{x'}, z\big) \big] \Big] \quad (4.12)$$

with θ_G, θ_E and θ_D representing the parameters of G, E and D, respectively.

We minimized $\mathcal{V}_{\text{BiGAN}}$ using stochastic gradient descent with Adam optimization, 64-sample mini-batch, and fixed learning rate of 2×10^{-4} for a total of $200{,}000$ epochs. Finally, we selected the encoder E corresponding to the last epoch.

4

Mean-RGB Baseline

We extracted the embedding e by averaging the pixel intensity across spatial dimensions from input RGB patches $x \in \mathbb{R}^{P \times P \times 3}$:

$$e^{(c)} = \frac{1}{P^2} \sum_{j=1}^{P} \sum_{k=1}^{P} x^{(j,k,c)} \quad (4.13)$$

with c indexing the three RGB color channels, and j and k indexing the two spatial dimensions.

Supervised-tumor Baseline

We trained an encoder E identical to the one used in the VAE model, followed by a dense layer with 256 units, BN and LRA; and finalized by a single sigmoid unit.

We minimized the binary cross-entropy loss using stochastic gradient descent with Adam optimization and 64-sample mini-batch, decreasing the learning rate by a factor of 10 starting from 1×10^{-2} every time the validation classification accuracy plateaued until 1×10^{-5}. Finally, we selected the encoder E corresponding to the model with the highest validation classification accuracy.

Supervised-tissue Baseline

We trained an encoder E identical to the one used in the VAE model, followed by a dense layer with 256 units, BN and LRA; and finalized by a softmax layer with nine units.

We minimized the categorical cross-entropy loss using stochastic gradient descent with Adam optimization and 64-sample mini-batch, decreasing the learning rate by a factor of 10 starting from 1×10^{-2} every time the validation classification accuracy plateaued until 1×10^{-5}. Finally, we selected the encoder E corresponding to the model with the highest validation classification accuracy.

4.8.2 Encoders for Synthetic Data

We modified the network architectures to account for the smaller patch size selected in the synthetic dataset. We used grayscale patches $x \in \mathbb{R}^{P \times P \times 1}$ with $P = 9$ and $C = 16$ unless stated otherwise. Note that we did not modify the training protocol, e.g. the learning rate schedule.

The architecture of the encoder E used in *VAE*, *Contrastive* and *Supervised* consisted of 2 layers of strided convolutions with 32 and 64 3×3 filters, BN and LRA; followed by a dense layer with 64 units, BN and LRA; and a linear dense layer with C units.

The architecture of the decoder D used in *VAE* consisted of a dense layer with 2048 units, BN and LRA, eventually reshaped to $(4 \times 4 \times 128)$; followed by 2 upsampling layers, each composed of a pair of nearest-neighbor upsampling and a convolutional operation[106], with 64 and 32 3×3 filters, BN and LRA; finalized with a convolutional layer with 1 3×3 filters and tanh activation.

With respect to *BiGAN*, the architecture of the encoder E consisted of 2 layers of strided convolutions with 32 3×3 filters, BN and LRA; followed by a linear dense layer with C units. The architecture of the generator G consisted of a dense layer with 512 units, BN and LRA, eventually reshaped to $(4 \times 4 \times 32)$; followed by 2 upsampling layers, each composed of a pair of nearest-neighbor upsampling and a convolutional operation[106], with 32 3×3 filters, BN and LRA; finalized with a convolutional layer with 1 3×3 filters and tanh activation. The discriminator D had two inputs, a low-dimensional vector and an image. The image was fed through a network with an architecture equal to E but different weights, and the resulting em-

bedding vector concatenated to the input latent variable. This concatenation layer was followed by two dense layers with 128 units, LRA and dropout (0.5 factor); finalized with a sigmoid unit.

4.8.3 Experiments

Patch-level Classification

On top of each encoder with frozen weights, we trained an MLP consisting of a dense layer with 256 units, BN and LRA; followed by either a single sigmoid unit or a softmax layer with nine units, respectively for each classification task.

We minimized the cross-entropy loss using stochastic gradient descent with Adam optimization and 64-sample mini-batch, decreasing the learning rate by a factor of 10 starting from 1×10^{-2} every time the validation classification accuracy plateaued until 1×10^{-5}. Finally, we selected the model with the highest validation classification accuracy.

Image-level Classification and Regression

We designed a CNN architecture consisting of 8 layers of strided depthwise separable convolutions[46] with 128 3×3 filters, BN, LRA, feature-wise 20% dropout, L2 regularization with 1×10^{-5} coefficient, and stride of 2 except for the 7-th and 8-th layers with no stride; followed by a dense layer with 128 units, BN and LRA; and a final output unit, with linear or sigmoid activation for regression or classification tasks, respectively.

We trained the CNN using stochastic gradient descent with Adam optimization and 16-sample mini-batch, decreasing the learning rate by a factor of 10 starting from 1×10^{-2} every time the validation metric plateaued until 1×10^{-5}. We minimized MSE for regression, and maximized binary cross-entropy for binary classification.

Visualizing where the information is located

Given a compressed gigapixel image ω', its associated image-level label y and a trained CNN S, we performed a forward pass over ω', producing a set of J intermediate three-dimensional feature volumes $f_j^{(k)}$, with j and k indicating the j-th and

k-th convolutional layer and feature map, respectively. Additionally, we computed the gradients of the feature volume $f_j^{(k)}$ with respect to the class output y, for a fixed convolutional layer. We averaged the gradients across the spatial dimensions and obtained a set of gradient coefficients $\gamma_j^{(k)}$ indicating how relevant each feature map was for the desired output y. Finally, we performed a weighted sum of the feature maps $f_j^{(k)}$ using the gradient coefficients $\gamma_j^{(k)}$:

$$h^{(k)} = \sum_{j=1}^{J} f_j^{(k)} \gamma_j^{(k)} \tag{4.14}$$

What we obtained was a two-dimensional heatmap $h^{(k)}$ that highlighted the regions of ω' that were more relevant for S to predict y. In order to maximize the resolution of the generated heatmap, we selected $k = 1$.

Grad-CAM heatmaps for Camelyon16 and TUPAC16 can be found here: `https://drive.google.com/drive/folders/16E-06rFbGab6-pXfjpo9vXjBVyUQKUuc`

Extending neural image compression with supervised multitask learning

5

Authors: David Tellez, Diederik Höppener, Cornelis Verhoef, Dirk Grünhagen, Pieter Nierop, Michal Drozdzal, Jeroen van der Laak, Francesco Ciompi

Original title: Extending unsupervised neural image compression with supervised multitask learning

Published in: International Conference on Medical Imaging with Deep Learning 2020

Abstract

We focus on the problem of training convolutional neural networks on gigapixel histopathology images to predict image-level targets. For this purpose, we extend Neural Image Compression (NIC), an image compression framework that reduces the dimensionality of these images using an encoder network trained unsupervisedly. We propose to train this encoder using supervised multitask learning (MTL) instead.

We applied the proposed MTL NIC to two histopathology datasets and three tasks. First, we obtained state-of-the-art results in the Tumor Proliferation Assessment Challenge of 2016 (TUPAC16). Second, we successfully classified histopathological growth patterns in images with colorectal liver metastasis (CLM). Third, we predicted patient risk of death by learning directly from overall survival in the same CLM data.

Our experimental results suggest that the representations learned by the MTL objective are: (1) highly specific, due to the supervised training signal, and (2) transferable, since the same features perform well across different tasks. Additionally, we trained multiple encoders with different training objectives, e.g. unsupervised and variants of MTL, and observed a positive correlation between the number of tasks in MTL and the system performance on the TUPAC16 dataset.

5.1 Introduction

Pathologists examine whole-slide images (WSIs) to diagnose a wide variety of diseases and predict patient prognosis. These WSIs are gigapixel images of human tissue sections taken at very high resolution, i.e. subcellular detail. In order to perform WSI classification, pathologists incorporate visual features from the entire WSI at once. This task poses two main challenges to Computer Vision algorithms: first, processing images of gigapixel resolution at once is computationally extremely expensive; and, second, the low signal-to-noise ratio present in WSIs has been shown to limit the performance of these algorithms[142].

Several methods have been proposed to solve WSI classification. In multiple-instance learning, a WSI is decomposed into small high-resolution patches (bag of patches) that are weakly annotated using the image label[110–115,143,144]. However, these methods cannot exploit the relationship between patches, being unable to observe a more global context of the WSI. Reinforcement learning has been proposed as a solution to increase context[145–147]. Although these methods can integrate knowledge across patches, they suffer from other limitations, e.g. optimization difficulties and leaving large areas of the WSI unexplored. Moreover, other authors have proposed memory-efficient methodologies that enable convolutional neural networks (CNNs) to be trained with very large images[131,148]. However, CNNs struggle to perform well in tasks with very low signal-to-noise ratio like histopathology image analysis[142], requiring vast amount of data samples to work. Unfortunately, histopathological datasets rarely surpass the hundreds or thousands of data points[82,94], urging for more sample-efficient methods to perform WSI classification.

Neural Image Compression (NIC) is a recently proposed framework[149] that can drastically reduce the dimensionality of WSIs while retaining semantic information and suppressing noise (see Fig. 5.1). NIC divides each WSI into a set of high-resolution small patches that are independently compressed into embedding vectors using an *encoder*, i.e., a neural network trained unsupervisedly. Then, these vectors are arranged in a 2D grid following the spatial configuration of the original WSI. The result of this operation is a compressed representation of the entire WSI, where each vector corresponds to a patch in the WSI. Once all WSIs are compressed using NIC, a classifier, e.g. a CNN, is trained on the compressed WSI representations using the image-level labels as targets. NIC addresses the main challenges of WSI classification by reducing both the size and noise levels of WSIs, while allowing the CNN classifier to exploit global context.

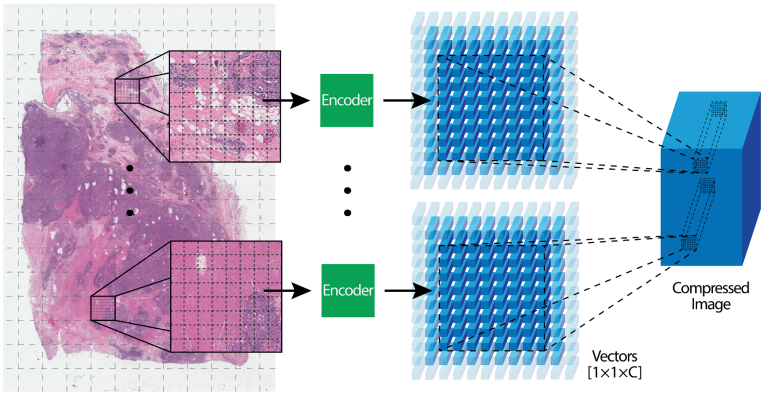

Figure 5.1: Neural Image Compression. Left: an entire gigapixel whole-slide image is read as a set of high-resolution patches using a uniform grid. Center: each of these patches is compressed into a low-dimensional embedding vector using a neural network, the encoder. Right: the embedding vectors are organized following the same spatial arrangement as in the original whole-slide image.

A crucial factor of the NIC method is the encoder model. This neural network is responsible for suppressing low-level pixel noise and spurious correlations, while identifying and extracting high-level discriminative features that could work well in a variety of downstream tasks and, as such, should be transferable across a number of histopathological tasks. To satisfy this condition, the original formulation of NIC suggested to use unsupervised methods to train the encoder, such as: variational autoencoders (VAE)[87], contrastive training[121–123], and adversarial feature learning[117,118]. Since networks trained with supervised signals are able to extract more specific feature representations than those using unsupervised loss terms[150], we hypothesize that combining multiple supervised goals during training could lead to superior and more generalizable features than using a single unsupervised task.

Therefore, we propose to introduce supervision in the training of the encoder and do so via supervised multitask learning[151]. Although, supervised and unsupervised multitask representation learning have shown promising results on multiple Computer Vision benchmarks[152,153], the usefulness of the learned representations for WSI compression is yet an unexplored research avenue. We propose a method that exploits and combines several supervision signals from four representative tasks in Computational Pathology: mitosis detection in breast, axillary lymph node tumor metastasis detection, prostate epithelium detection, and colorectal cancer tissue type

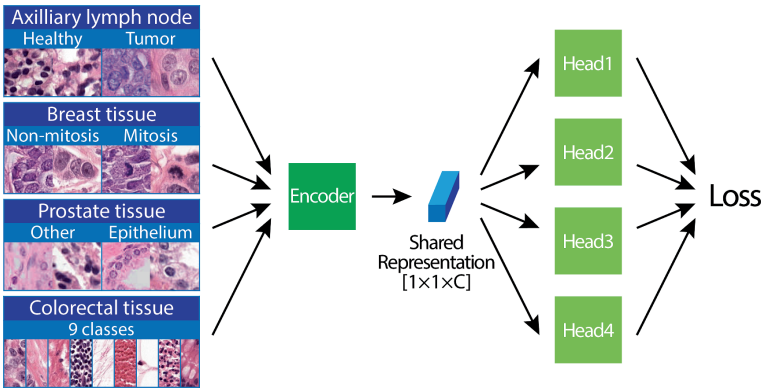

Figure 5.2: Supervised multitask learning framework. Left: the full model is trained to solve four different tasks simultaneously. Center: the encoder provides a shared embedded representation for the images of all the tasks. Right: the head models perform each of the four classification tasks independently from each other.

classification.

In this work, we trained image compression using the proposed multitask NIC and evaluated the obtained representations in two histopathology datasets that target image-level labels. First, modeling the speed of tumor growth in invasive breast cancer, included in the Tumor Proliferation Assessment Challenge 2016 (TUPAC16)[82]. Second, predicting histopathological growth patterns and the overall risk of death in patients with colorectal metastasis in the liver[154].

Our contributions can be summarized as follows:

- We improved NIC by training the encoder with supervised multitask learning. Experimental results suggest that embedding vectors were more discriminative and transferable, and adding more tasks to the multitask framework increased the performance of the method at WSI level.
- We obtained state-of-the-art performance predicting tumor proliferation speed in invasive breast cancer patients from the Tumor Proliferation Assessment Challenge, and classifying histopathological growth patterns in patients with colorectal liver metastasis.
- We successfully predicted patient risk of death by learning directly from overall survival in patients with colorectal liver metastasis, without the need for human intervention.

5.2 Materials

Multitask learning dataset. We selected multicenter data from four representative patch classification tasks in Computational Pathology (see Fig. 5.2), namely: mitosis detection in breast, axillary lymph node tumor metastasis detection, prostate epithelium detection, and colorectal cancer tissue type classification. A full description of this dataset is available in[155]. For each task, we selected 200000 patches of 64×64 pixels at $0.5\,\mu\mathrm{m/pixel}$ resolution with patch-level annotations. We distributed the number of patches across classes and medical centers uniformly, so that classes and centers were equally represented in the dataset, and reserved 20% of the samples for validation purposes (randomly selected).

TUPAC16 dataset. We used public WSIs from the Tumor Proliferation Assessment Challenge 2016 (TUPAC16)[82] to evaluate our method. This cohort consisted of 492 hematoxylin and eosin (H&E) training slides taken from patients with invasive breast cancer from The Cancer Genome Atlas[73]. The organizers of the Challenge provided a label for each patient that served as a proxy for tumor proliferation speed[74]. Additionally, the organizers also provided 321 test slides with no public labels available, that were used to perform a truly independent performance evaluation.

Colorectal liver metastasis dataset. This private cohort consisted of 363 patients that underwent colorectal liver metastasis resection at the Erasmus MC Cancer Institute (Rotterdam, the Netherlands) between 2000 and 2015[154]. A total of 1571 H&E stained slides were used in this work. These slides were scanned using a 3DHistech P1000 scanner at a spatial resolution of $0.25\,\mu\mathrm{m/pixel}$. Each slide was manually scored for the presence of histopathological growth patterns (HGP) following international guidelines[156,157]. A given slide was considered desmoplastic HGP (dHGP) if this was the only pattern observed, and non-desmoplastic HGP (non-dHGP) otherwise. The consideration of dHGP is generally associated with better prognosis (longer patient survival). In addition, overall survival was available for these patients, with a mean follow-up period of 3.4 years.

5.3 Methods

Supervised multitask learning. Our multitask learning architecture is built from two components. The first component, the encoder, is shared among the four tasks,

whereas the second part, the heads, consists of four multilayer perceptrons (MLPs) specialized in solving each task individually. Both the encoder and the four heads are trained to minimize the sum of the classification losses of the four tasks. By doing so, the encoder learns a vector representation that is optimized to produce high classification performance while being highly transferable across different tasks. Fig. 5.2 provides an overview of the method. The size of the embedded representation C is an hyperparameter of the method, by default set to $C = 128$ following the original implementation of NIC. We trained this model using images from the *multitask learning dataset* only.

The architecture of the encoder consisted of 4 strided convolutional layers with 128 3×3 filters, batch normalization, leaky-ReLU activation (LRA), and stride of 2; followed by a linear layer with C units. The head models were composed of a dense layer with 256 units, and LRA, with 10% dropout before and after this layer; and a final dense layer whose number of output units depended on the classification task (9 for the multiclass tissue classification, and 2 for the rest), followed by a softmax.

We trained the encoder and head models simultaneously by minimizing the average categorical cross-entropy across the four tasks. We used stochastic gradient descent with Adam optimization and a mini-batch of 128 samples (32 samples per task dataset), decreasing the learning rate by a factor of 10 starting from 1×10^{-3} every time the validation metric plateaued until 1×10^{-5}. During training, we used heavy morphological and color augmentation[155], increasing the model robustness to unseen data.

WSI classification. In order to train a CNN classifier on gigapixel WSIs and image-level labels, we followed the method described in the original NIC publication, with the exception of using the proposed multitask encoder instead of the unsupervised model. A detailed description of the CNN architecture and training details is available in the Appendix 5.7.1.

Learning from patient overall survival. Survival analysis constitutes a regression problem where a model is trained to predict a risk score for each patient that is proportional to their chances of experiencing the event of death. Each patient's WSI is associated with a record composed of two items: a follow-up period and a binary death-event variable. We used WSIs compressed with multitask NIC and overall survival data to train a CNN classifier to predict patient risk of death by maximizing the partial log-likelihood loss[158]. Intuitively, by optimizing this objective the CNN

Table 5.1: Predicting tumor proliferation speed in TUPAC16 Spearman corr. and 95% c.i.

Method	Training set	External test set
TUPAC16 top-3 [82]	-	0.503
TUPAC16 top-2 [82]	-	0.516
NIC unsupervised [149]	0.522	0.558 [0.5191, 0.5962]
Streaming CNNs [148]	-	0.570
TUPAC16 top-1 [82]	-	0.617
NIC multitask (proposed)	0.620	**0.632 [0.5966, 0.6641]**
TUPAC16 human-assisted (*) [82]	-	0.710

(*) Requires the intervention of an expert pathologist

classifier learned to assign high risk scores to those patients that died early. See Appendix 5.7.3 for a detailed description of this loss.

5.4 Experimental results

5.4.1 Training the multitask encoder

For the purpose of compressing WSIs in the TUPAC16 and liver datasets, we first trained a 4-task multitask encoder following the procedure described in the Methods section. We obtained the following validation accuracy scores at patch level: lymph node tumor classification (90.87%), mitosis classification (94.81%), prostate epithelium classification (86.48%), and colorectal 9-class classification (77.49%).

5.4.2 Predicting the speed of tumor proliferation (TUPAC16)

In this experiment, we used the previously trained encoder to compress the WSIs on the TUPAC16 training dataset; and trained four CNN regressors on top of these compressed WSIs using 4-fold cross-validation (3 folds for training, 1 for validation). For the test set, we compressed the WSIs similarly, then averaged the predictions of the four CNNs per sample, and submitted the results to the Challenge organizers for independent evaluation (the labels of the test set are not public). Our proposed method achieved state-of-the-art results on the leaderboard of the TUPAC16 Challenge for automatic methods (see Tab. 5.1), demonstrating the effectiveness of using multitask learning in combination with NIC to predict image-level labels from WSIs.

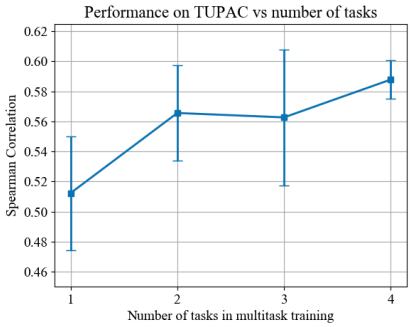

Figure 5.3: Relationship between the number of tasks used to train the multitask encoder and the performance of the CNN regressor trained on TUPAC16 (mean and std Spearman corr).

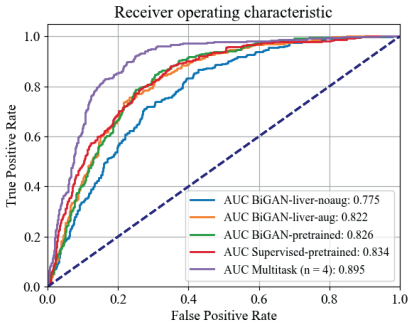

Figure 5.4: Predicting the presence of desmoplastic HGP in colorectal liver metastasis WSIs. Different encoding strategies are compared using the area under the ROC.

5.4.3 Image-level performance vs. number of tasks used in multitask training

The goal of the following experiment was to study the relationship between the number of tasks used to train the multitask encoder and the performance of the CNN regressor trained at image level, i.e. using TUPAC16 WSIs. First, we trained a set of encoders varying the number of tasks included during multitask training: 4 encoders using 1 task (only lymph, only mitosis, etc.), 6 encoders using 2 tasks (lymph+mitosis, lymph+prostate, etc.), 4 encoders using 3 tasks (lymph+mitosis-+prostate, lymph+mitosis+colorectal, etc.), and 1 encoder using 4 tasks. Second, we compressed the TUPAC16 training dataset using each of the 15 previously trained encoders, and trained CNN regressors on them using 4-fold cross-validation as before in order to obtain an unbiased prediction for each training sample. Due to the large computational resources required to perform these steps, we used a reduced code size of $C = 16$.

We measured the Spearman correlation between the predictions of our system and the image-level labels, and averaged the results by the number of tasks (see Fig. 5.3). Note that we repeated the 4-task experiment four times with random weight initialization to obtain a more robust performance estimate. All the performance metrics are summarized in the Appendix 5.7.2. We observed that increasing the number of tasks during multitask training produced a higher and more robust performance at

Table 5.2: Spearman correlation between task inclusion during multitask training, and performance at image level in TUPAC16

	Lymph	Mitosis	Prostate	Colorectal
Correlation	0.319	0.033	0.077	0.824

image level. However, the large variance obtained in some cases (2 and 3 tasks) suggests that task selection might play an important role in the performance of multitask NIC.

Additionally, we measured the Spearman correlation between a binary variable describing whether a task was included during multitask training or not, and the performance of the system at image level. The results of this analysis are presented in Tab. 5.2. We found a positive correlation between the inclusion of the colorectal task and the global performance at image level. This result suggests that this dataset might be more valuable for feature extraction purposes than the rest. We recognize this task to be the most complex of all, requiring the encoder to extract robust features to accurately solve the classification problem. We hypothesize that multitask training can benefit from the highly specific features required to solve difficult classification tasks like this one.

5.4.4 Predicting patient risk of death in colorectal liver metastasis

Desmoplastic HGP manual annotations. We compressed all the liver WSIs using the multitask encoder introduced before in Sec. 5.4.1, and trained a CNN classifier to distinguish between dHGP or non-dHGP type on the compressed WSIs using 4-fold cross-validation (2 folds for training, 1 for validation, and 1 for testing). We measured the area under the ROC (AUC) on all the test samples to quantify performance.

In addition, we considered the predicted probability of dHGP as a proxy for the patient risk of death, and used the Kaplan-Meier (KM) estimator to model survival curves for two groups of patients, low and high risk, divided by the median predicted risk score.

For the dHGP classification task, we obtained an AUC of 0.895. Regarding the prognostic power of these predictions, results in Fig. 5.5 showed that our system could divide the population into two risk categories with high significance ($p < 0.001$).

Overall survival records. We trained a CNN classifier on the same compressed liver

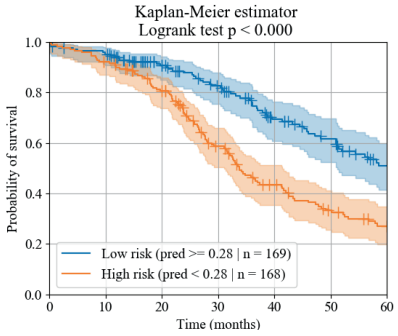

Figure 5.5: Predicting patient risk of death in colorectal liver metastasis WSIs. Learning from annotated HGP status.

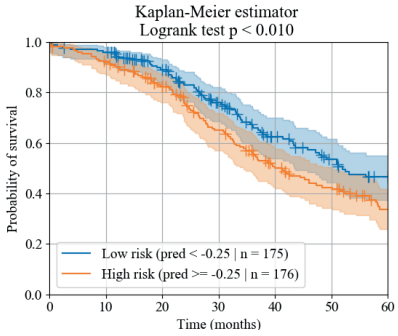

Figure 5.6: Predicting patient risk of death in colorectal liver metastasis WSIs. Learning from overall survival records.

WSIs to predict patient risk of death learning directly from overall survival data (loss described in Sec. 5.3 under *Learning from patient overall survival*). As before, we used the KM estimator to assess the prognostic power of the classifier's predictions.

We found that this model was able to learn directly from overall survival data, dividing the population into two risk categories ($p < 0.01$), see Fig. 5.6. Note that no manual annotation was required on the colorectal liver metastasis dataset to perform this experiment, only patient records.

5

5.4.5 Comparing unsupervised and supervised encoders

We repeated the experiment described in Sec. 5.4.4 under *Desmoplastic HGP manual annotations* using different encoding options. In particular, we compressed the liver WSIs using several unsupervised and supervised encoding methods, and subsequently trained a CNN classifier to distinguish between dHGP and non-dHGP status.

We selected several encoders from the original NIC publication[149] as baselines for the comparison: the unsupervised bidirectional generative adversarial network (*BiGAN-pretrained*) and the supervised network trained for lymph node tumor classification (*Supervised-pretrained*). Additionally, we trained two BiGAN encoders using patches extracted from the liver WSIs; in one case applying no augmentation during training (*BiGAN-liver-noaug*), and using heavy color augmentation in the other one

(*BiGAN-liver-aug*). Finally, we also compared our proposed 4-task multitask encoder (*Multitask* $n = 4$).

Evidence in Fig. 5.4 highlighted three main results. First, heavy color augmentation played an important role in improving the features extracted by the encoder. Second, there seemed to be no difference between unsupervised and one-task supervised methods, trained on liver or any other organ. Three, multitask training substantially improved the overall performance of the system, obtaining the best classification AUC score of all tested methods (0.895).

5.5 Discussion

In this study, we extended Neural Image Compression[149] by training the encoder with a supervised multitask learning approach. We trained the encoder to solve four classification tasks in Computational Pathology simultaneously, and used this model to perform the gigapixel image compression. First, supervised multitask training was key to obtaining a high performance at image level, surpassing unsupervised techniques. We found that increasing the number of tasks used to train the encoder was directly proportional to the system performance. Second, we obtained state-of-the-art results in predicting both the speed of tumor proliferation in invasive breast cancer (TUPAC16 Challenge), and HGP status in colorectal liver metastasis classification. These results in real-world tasks showcased the flexibility of multitask NIC as a method to empower WSI classification. Third, we used the proposed system to assess patient risk of death by learning directly from overall survival data, i.e. without human intervention. By doing so, we enabled the CNN classifier to work as an effective biomarker discovery tool for liver metastasis, moving beyond human knowledge rather than mimicking pathologists.

We acknowledge the main limitation of the proposed method to be a lack of straightforward criteria on how to expand the number and type of tasks used during multitask training, i.e. which tasks to select and include in the multitask loss function. We selected four representative tasks performed in the clinic with high-quality patch-level annotations. However, our results suggest that the WSI classifier might be sensitive to this choice. Careful weighting of multitask objectives and optimizing which tasks should be learned together is a matter of study in recent publications in the field[159–161]. Future work should focus on conducting a more detailed evaluation on how to select these patch-level tasks, combining multiple objectives optimally, and

including unsupervised or weakly-annotated data in the multitask loss.

5.6 Acknowledgement

This study was supported by a Junior Researcher grant from the Radboud Institute of Health Sciences (RIHS), Nijmegen, The Netherlands; a grant from the Dutch Cancer Society (KUN 2015-7970); and another grant from the Dutch Cancer Society and the Alpe d'HuZes fund (KUN 2014-7032); this project has also been partially funded by the European Union's Horizon 2020 research and innovation programme under grant agreement No 825292. The authors would like to thank Dr. Mitko Veta for evaluating our predictions in the test set of the TUPAC16 Challenge[82] dataset; and the developers of Keras[141], the open source tool that we used to run our deep learning experiments.

5.7 Appendix

5.7.1 Architecture and training details of the image-level CNN

The complete CNN architecture consisted of 8 convolutional layers using strided depthwise separable convolutions with 128 3×3 filters, batch normalization (BN), leaky-ReLU activation (LRA), L2 regularization with 1×10^{-5} coefficient, feature-wise 20% dropout, and stride of 2 except for the 7-th and 8-th layers with no stride; followed by a dense layer with 128 units, BN and LRA; and a final layer that depended on the application: a softmax dense layer for classification problems, and a linear output unit for regression tasks.

We trained the CNN using stochastic gradient descent with Adam optimization and 16-sample mini-batch, decreasing the learning rate by a factor of 10 starting from 1×10^{-2} every time the validation metric plateaued until 1×10^{-5}. We minimized mean squared error for regression (TUPAC16), cross-entropy for classification (dHGP vs non-dHGP), and partial log-likelihood for patient risk prediction (targeting overall survival).

Table 5.3: Predicting tumor proliferation speed in TUPAC16 (Spearman corr.) depending on which task was included during multitask training

Lymph	Mitosis	Prostate	Colorectal	Correlation
No	No	No	Yes	0.563
No	No	Yes	No	0.515
No	Yes	No	No	0.473
Yes	No	No	No	0.498
No	No	Yes	Yes	0.584
No	Yes	No	Yes	0.573
No	Yes	Yes	No	0.557
Yes	No	No	Yes	0.613
Yes	No	Yes	No	0.548
Yes	Yes	No	No	0.520
No	Yes	Yes	Yes	0.549
Yes	No	Yes	Yes	0.592
Yes	Yes	No	Yes	0.605
Yes	Yes	Yes	No	0.505
Yes	Yes	Yes	Yes	0.569
Yes	Yes	Yes	Yes	0.594
Yes	Yes	Yes	Yes	0.597
Yes	Yes	Yes	Yes	0.591

5.7.2 Multitask training experiments

In Sec. 5.4.3, we experimented varying the number of tasks included during multitask training. We trained a set of encoders that were subsequently used to solve the TUPAC16 task. In Tab. 5.3, we show the performance obtained with each encoder.

5.7.3 Learning from overall survival data

Survival analysis constitutes a regression problem where a model is trained to predict a risk score for each patient that is proportional to their chances of experiencing a given event, in this case death. Each patient's WSI is associated with a label composed of two items: a follow-up period t indicating the number of months that the patient has been enrolled in the study; and a binary variable e that describes the event of whether the patient actually died or not. If a patient is event-free until her last follow-up, i.e. did not die, that sample is considered to be *censored* since we do not know when the event could take place in the future. The following introduces a

loss function that enables a CNN to regress the patient's risk of death by exploiting censored and uncensored data.

The Cox's proportional hazards model[162] is the most widely used method to find the relationship between censored data and its covariates x. It models the hazard function $h(t, x)$ (probability of death at a given time) as a product between a time-dependent baseline $h_0(t)$ that is common to all patients, and a term proportional to the covariates weighted by learned coefficients β:

$$h(t, x) = h_0(t) \exp(\beta x). \tag{5.1}$$

We can estimate the optimal coefficients $\hat{\beta}$ by maximizing the logarithm of the partial likelihood:

$$\hat{\beta} = \arg\max \log \prod_{i \in D} \frac{\exp \beta x_i}{\sum_{j \in R_i} \exp \beta x_j} = \arg\max \sum_{i \in D} \left(\beta x_i - \log \sum_{j \in R_i} \exp \beta x_j \right), \tag{5.2}$$

where D corresponds to the set of patients whose death is recorded (uncensored), and R_i is the set of patients that did not experienced the event before patient i, i.e. survived longer than patient i.

In this study, the covariate vector x is a compressed WSI γ representing a patient. Instead of weighting each pixel with β coefficients to produce a risk score, we parameterize this transformation with a CNN as $f(\theta, \gamma)$, with θ representing the trainable parameters of the neural network[158]:

$$\hat{\theta} = \arg\max \sum_{i \in D} \left(f(\theta, \gamma_i) - \log \sum_{j \in R_i} \exp f(\theta, \gamma_j) \right). \tag{5.3}$$

Intuitively, by maximizing the previous term we enforce the CNN to learn a certain solution $\hat{\theta}$ that assigns high risk scores $f(\theta, \gamma)$ to those patients $i \in D$ that died early in comparison to those patients in each i's risk set R_i that survived longer, and thus deserve a lower risk score.

General discussion

6

6.1 Introduction

In this work, we investigated how deep learning based methodologies could be developed and applied in histopathology image analysis with the long-term goal of improving breast cancer prognosis. We explored a wide variety of problems in Computational Pathology, and produced innovative technical solutions to tackle them. In this chapter, we analyze the results and conclusions obtained in the studies conducted within this thesis, interpreting their impact in the field and providing an outlook to the future.

6.2 Automating mitosis detection in breast cancer

One of the cornerstones of breast cancer grading, i.e. the histopathological assessment of tumor aggressiveness and prognosis, is mitosis counting. Determining the number of cells undergoing mitosis within the tumor lesion has been shown to be a reliable and independent prognostic biomarker[11,12]. Therefore, in Chapter 2 we investigated how to create a deep learning based mitosis detector that could provide a reliable and robust measure of mitosis counting automatically, and the challenges involved in such endeavour.

We identified several major issues holding back the development of an excellent mitosis detector. First, we noticed a high inter-observer variability when experienced pathologists were asked to annotate mitotic figures in entire whole-slide images (WSIs). This observation pointed out that mitosis detection is a hard problem even for human observers. Thus, manually identifying and annotating a large set of mitotic figures to create a reference standard for training was very expensive in the best case, and often unfeasible within the time frame and budget of our research. Moreover, given the observer variability of identifying mitoses, a *real* reference standard can never be established, necessitating a fallback to, e.g., consensus or majority voting. To overcome this limitation and effectively scale-up the reference standard generation process, we proposed a technique based on immunohistochemical (IHC) restaining. This method consisted in restaining the same hematoxylin and eosin (H&E) slide with an IHC stain that highlighted our biomarker in the slides, in our case mitotic figures. By having two images representing the same tissue, we could detect tens of thousands of mitotic figures in IHC with ease, then transfer them to H&E automatically. This methodology can be used in any other pathology application where the reference standard can be detected with an IHC staining. Since the

publication of our work, other researchers have successfully used this technique to generate a reference standard for epithelial cells in prostate tissue[97], showcasing the flexibility of our approach.

A second major issue impeding robust and reliable automatic mitosis detection was inter-center stain variation. Generally speaking, deep learning models trained with images from a given center often underperform when applied to images created in different laboratories. We investigated the reasons behind this lack of generalization and hypothesized that none of these models were trained to become truly invariant to changes in stain appearance. We argued that the most widespread technique to palliate this effect, stain normalization, was only exacerbating the problem by limiting the input color space to that of a single center. We proposed to use heavy data augmentation during training to actively enforce the model to ignore stain variations, so that it would learn discriminative features that were independent from stain characteristics. Furthermore, we designed a specific color augmentation protocol based on H&E channel decomposition, producing realistic and varied stain appearances during the training phase. This specific H&E augmentation improved the performance of the mitosis detector, and we believe that this data augmentation protocol should be the default choice when training deep learning models in any histopathology application. Since we published this work, numerous researchers have acknowledged, embraced or extended our augmentation procedure[163–179], signaling a clear change from classic stain normalization towards the use of stain augmentation during training.

Our research produced an accurate, robust and stain-invariant mitosis detector that identifies mitotic figures throughout entire WSIs, locating the hotspot, i.e. the densest region, and providing the number of detections within the hotspot. Since its conception, this tool has been successfully applied to tens of thousands of images from dozens of centers, even from different organs than breast. Follow-up studies to evaluate the performance of this mitosis detector in the clinical setting have been conducted in recent years[180], concluding that automated quantification of the mitotic score was *"feasible without introducing additional bias or variability"*. Following this validation, the same algorithm has been used to accelerate mitosis counting, allowing researchers to analyze the prognostic information of mitotic scoring at an unprecedented scale[181]. As more tasks are automatized in pathology, we believe that more avenues for research would be enabled, allowing Computational Pathology to take a leading and transformative role in the clinical setting.

6.3 Addressing inter-center stain variation

Inter-center stain variation is caused by multiple factors, generally explained by the staining discrepancies between different pathology laboratories and scanning vendors. Typically, models trained with images from a given center tend to underperform in images originated in different labs due to a variety of perturbations related to color shifts. In Chapter 2, we identified inter-center stain variation as a major limiting factor for the performance of deep learning based systems in Computational Pathology, and hinted at a potential solution to the problem. In Chapter 3, however, we dived deeper into this issue by analyzing and comparing several methodologies that were known to produce more robust methods. Moreover, we evaluated these methodologies across different representative tasks in order to extract general conclusions and recommendations for the field of Computational Pathology.

Historically, stain normalization has been the most used method to prevent inter-center stain variation. Transforming the stain of all images, i.e. training and test, to that of a common template prevents the model to perceive any stain deviations and thus performs as expected. However, these methods act as a single-point of failure and any imperfection in the normalization process can lead to suboptimal overall performance. Our experiments showed that using stain normalization alone was not sufficient to address the inter-center stain variation problem when tested across multiple tasks. Moreover, we argued that simulating inter-lab domain shift was a more effective task than hand-engineering a routine to perform stain normalization. Following this approach, we proposed a deep learning based method to transform images from any source to look like those in the training set by removing simulated stain variations. Our results showed that this and other network-based methods effectively outperformed more classical approaches, establishing the type of normalization methods recommended for future use.

Since the advent of deep learning, data augmentation has played a major role in model robustness, and it has emerged as an indispensable tool to fight inter-center stain variation. By simulating a wide spectrum of stain appearance during training, the model learns to ignore features that depend on stain characteristics, effectively becoming invariant to inter-center stain variations. We found that heavy stain augmentation was sufficient to obtain top classification performance across the evaluated tasks. This and previous results have provoked a paradigm shift from using stain normalization alone to embracing heavy data augmentation, and have influenced a number of researchers to acknowledge and adopt this methodology in their

work[163–179]. We believe that data augmentation has become a critical building block of every Computational Pathology training pipeline, with a strong emphasis on stain and color variations.

6.4 Gigapixel image classification targeting patient-level labels

A major research aim within the scope of this thesis has been the development of a deep learning based methodology that can make predictions from full-sized WSIs. More specifically, we investigated models that could ingest an entire gigapixel WSI and predict an image-level label by integrating information from both the global and the local context in the image. Previous approaches have neglected this idea of analyzing full images and have simplified the process by targeting local features only, e.g. by annotating pixel-level patterns such as tumor cells or mitotic figures. We have recognized the importance of patient prognosis as the true reference standard for patient and disease outcome, and developed a method that was able to learn directly from these patient records. For example, a classical computer-aided diagnostic pipeline would first detect tumor cells in the WSI (task 1), then move onto providing measurements and other hand-engineered features on these detected lesions (task 2), to finally predict patient diagnosis or prognosis based on a pre-established clinical guideline (task 3). Our proposal was to develop a deep learning based method to predict patient diagnosis or prognosis directly from the raw input data (WSIs), allowing this model to discover and exploit patterns related to disease outcome without human intervention.

When attempting to model the relation between WSIs and image-level targets, two main issues arose. First, the gigapixel resolution of these images imposed an unassumable computational burden on our hardware budget. Processing billions of pixels with a single GPU, or even a few of them, resulted in extremely slow processing times that made experiments unfeasible. Second, even if these images could fit into memory, training a model with the input dimensionality of a gigapixel image would require a number of data samples that vastly surpassed what was available at the time (a few thousand images in the best case). Therefore, in Chapter 4 we designed a method that could drastically reduce the size of the input image while maintaining the high-level semantic information present in the raw input intact.

We introduced Neural Image Compression (NIC), an algorithm that divides the process of predicting image-level targets from gigapixel images in two phases: the *compression* and the *training* phase. During the *compression* phase, an encoder network reduces the size of the input image by extracting feature vectors from local image patches. Subsequently, the *training* phase trains a CNN on top of these compressed images targeting an image-level label using standard deep learning tools. NIC is based on the idea that most of the pixel-level patterns present in WSIs can be highly compressed using a neural network. Furthermore, this neural network compressor can be trained using unlabeled data which is plentiful and available at almost no cost. What we found in our research was that neural networks were very effective at extracting high-level features and delivered unprecedented levels of image compression, while still maintaining key information about the raw input. Moreover, our research pointed out that we could train the encoder using images that were different from those used in the final image-level application. This piece of evidence suggested that we could use a fixed encoder network to compress any kind of WSI, regardless of the application or target that we would like to predict. This finding decoupled the *compression* phase from the *training* phase, enabling each phase to be performed by different actors. Future applications could exploit this idea by, e.g., providing compression of WSIs as an independent online service, with different encoding options and configurable parameters.

The ability to exploit global and local features simultaneously is a key competitive advantage of our method, and we believe it to be the main driver of prediction performance. We envision future extensions of NIC in several areas. First, improving the encoding procedure by means of using more effective unsupervised learning mechanisms, stronger supervision, or innovative representation learning methods. Second, designing more sample-efficient models that exploit morphological patterns in compressed images more effectively. Third, extending the range of applications beyond classification and regression tasks, e.g., to perform semantic segmentation at full image-level scale. This approach could have the added benefit of accessing a much broader image context when compared to regular patch-by-patch segmentation. More generally, since NIC is a target- and task-agnostic method to deal with WSIs, compressed images could be used to represent the patient's histopathological status, which combined together with the patient's health records and history could enable a more data-driven description of each patient ready to be exploited by a variety of machine learning methodologies.

6.5 Predicting patient prognosis from whole-slide images

Patient survival is the most fundamental and unbiased form of reference standard that a machine learning model can learn from. With almost no human intervention, it provides the true gold standard about the health status of a given patient. Although deep learning models can exploit this kind of reference standard, it comes with its own drawbacks. Survival data is known to be very noisy and scarce, since patients might die for unrelated causes and might drop out clinical studies without explanation. In order to discover discriminative patterns that describe the future outcome of the disease under consideration, models exploiting these labels must be very sample efficient. In Chapter 5, we enhanced the efficiency of NIC by incorporating supervised multitask learning, a method that improved the encoder network using several supervised signals during training, and achieved an unprecedented performance in several tasks.

Supervised multitask learning worked by training a regular neural network to solve several classification tasks simultaneously, that is, the parameters of such model were optimized to jointly minimize the losses of multiple classification goals. Our proposal involved four of the most common histopathology tasks: detection of mitotic figures, detection of tumor cells, detection of epithelial cells, and tissue-type multiclass classification. By learning to solve these four tasks simultaneously, the encoder was trained to recognize highly discriminative and specific features, while being actively encouraged to learn a feature representation that was transferable across multiple domains and tasks. This combination of specificity and transferability drastically improved the quality of the features present in each of the compressed WSIs after the *compression* phase, easing the task of the CNN applied to them during the *training* phase.

In Chapter 5, we showed that improved NIC could be used to solve a clinically relevant problem with state-of-the-art performance, by predicting the speed of tumor proliferation in invasive breast cancer (TUPAC 2016 Challenge). We believe that this milestone marks the beginning of a new framework methodology in Computational Pathology. For the first time, we can train any standard deep learning based method on WSIs and image-level labels, achieving state-of-the-art results. The flexibility and performance of multitask NIC has positioned itself as the default choice for this kind of image-level application, justifying the creation of even larger histopathology datasets with tens or hundreds of thousands patients per cohort.

Multitask NIC produced compressed images that were much more descriptive than any previous attempts, effectively reducing the sample-efficiency requirements of the image-level problem. Moreover, we obtained remarkable results when we used this method to estimate patient risk of death by learning from overall survival data directly, without human intervention at all. This result suggested that NIC can be used as a biomarker discovery tool, i.e., this methodology can potentially find visual features linked to the disease and eventual death of a given patient. We believe this to be a revolutionary idea in the field of Computational Pathology since these systems could evolve from being simple automations of repetitive tasks, to constituting a fundamental tool in medical research and knowledge discovery. Our research has drawn substantial attention in the Computer Vision community with Facebook AI Research writing a blog post entry about it[182], and the Computer Vision News Magazine featuring our research as a runner-up for best-paper award in the MIDL20 conference[183].

6.6 Future outlook

It is important to recognize that scientists in the field of Computational Pathology are not only automating tedious and repetitive tasks but revolutionizing how the entire science of pathology is performed. Pathologists have now access to tools that enable them to conduct research that was not possible before due to unacceptable costs in terms of human and material resources, lengthy development time, or high inter-observer variability. Demonstrating this trend, we developed a tool to perform automatic mitosis counting which allowed not only to detect mitotic figures seamlessly, but to answer completely new research questions. For example, Maschenka Balkenhol et al. studied whether mitotic counting could be an effective prognostic biomarker in triple negative breast cancer patients by analyzing hundreds of thousands of mitotic figures in hundreds of patients[181]. This piece of research would not have been possible without an efficient method to perform mitosis detection in entire whole-slide images. As the technology advances, we predict that Computational Pathology solutions will deliver even more value beyond solving repetitive tasks, and accelerate science in the field of clinical pathology.

The vast majority of the deep learning based solutions currently developed for Computational Pathology are focused on automating clinical tasks that pathologists consider to be highly repetitive and time-consuming. Although this kind of methods

will continue to exist and expand in the near term, we believe that the bulk of the future growth will be concentrated around ideas and methodologies that learn directly from image-level targets. Pathology laboratories have no shortage of WSIs and patient records in their archives, however, they cannot afford expensive manual annotations at pixel level for all of them. By scanning these images and using solely patient records as training labels, research teams could scale-up their experiments to the hundreds of thousands of patients within their resource budget in a timely fashion. Moreover, we envision a future where multiple WSIs per patient are considered and analyzed together by deep learning systems, e.g. several H&E and IHC consecutive slides from the same tissue block. Since NIC offers a translation tool from raw pixel space to a more expressive feature space, all these different image modalities can be easily exploited together, even combined with other sources of information like clinical records, anatomical information, or other medical modalities such as radiology (e.g. CT scans and X-ray images) or genetics. This method of learning from vast amounts of multiple heterogeneous forms of input data and patient record targets is unprecedented and constitutes one of the most comprehensive approaches in machine learning applied to healthcare. We believe that it would lead to a revolution in knowledge and biomarker discovery, especially in widespread diseases like cancer.

6.7 Conclusion

This thesis describes the research work conducted around several fundamental problems in Computational Pathology. In Chapter 2, we proposed a method to scale up the creation of pixel-level annotations by combining IHC staining with WSI registration. We leveraged this process to create a reference standard for mitosis detection of unprecedented size, and trained a robust mitosis detector that has been used in numerous clinical studies since then. In Chapter 3, we studied the problem of inter-center stain variation across multiple tasks, and concluded that heavy color augmentation was key to develop stain-invariant convolutional neural networks. We proposed a data augmentation procedure that has been used across many Computational Pathology applications so far. In Chapter 4, we proposed NIC, a novel method that drastically reduced the size of WSIs so that they could be used to train models targeting image-level labels directly. We further improved the method in Chapter 5 by training the encoder network with supervised multitask learning, which resulted in compressed WSIs whose features were more discriminative and transferable across tasks than ever before. We obtained state-of-the-art performance when

predicting the speed of tumor proliferation in invasive breast cancer. Furthermore, we trained a model to predict patient risk of death by learning directly from overall survival records, without manual intervention. According to these results, we believe that NIC constitutes a revolutionary knowledge and biomarker discovery tool for the field, and will enable deep learning models to learn from a combination of medical modalities such as histopathology, patient history, radiology and genetics, with almost no human intervention.

Summary

In **Chapter 1**, we provided an introduction to the fundamental ideas of this thesis and provided an extensive background to the reader. More specifically, we described the field of Computational Pathology, and its connection with histopathology and artificial intelligence. Histopathology is the science that studies the microscopic structure of human tissue in order to understand the mechanisms of disease. With the advent of information technology, the field is undergoing a profound process of digitization. With the development of new image analysis tools, pathologists are starting to use more and more of these algorithms in both research and clinical routine. A particular family of image analysis algorithms, deep learning, has dominated the entire field of Computer Vision for the past few years due to its effectiveness across multiple tasks. In this chapter, we described the basis to understand the Computational Pathology revolution and presented the goals of this thesis:

- To address fundamental challenges in Computational Pathology such as: generating pixel-level annotations, inter-center stain variation, and processing entire whole-slide images.
- To automate a core component of breast cancer grading: performing mitosis detection at scale, and deriving actionable insights for the pathologists.
- To design a novel method that can perform gigapixel whole-slide image classification, with the goal of learning from patient overall survival.

In **Chapter 2**, we proposed a method that can automatically perform robust mitosis detection throughout entire H&E whole-slide images from breast cancer patients. We presented three main contributions to the field. First, we developed a method that exploits H&E stain and immunohistochemistry images to create one of the largest training sets for mitosis detection to date. Second, we designed a procedure to ensure that the mitosis classifier was robust to stain variations. Third, we proposed a protocol to train the detector to maximize classification performance using hard negative mining, ensembling, and knowledge distillation. The proposed mitosis detection solution has been used in thousands of patients from dozens of centers around the world, in several other papers after our publication, and it has proven to be a valuable research tool within the medical domain.

In **Chapter 3**, we addressed the problem of intra- and inter-center stain variation, and its effect in performance on convolutional neural networks for several Computational Pathology applications. We analyzed two approaches to solve this problem: stain augmentation and stain normalization. In both cases, we systematically compared several of the most used techniques and some of our own proposals. After

performing hundreds of experiments, we concluded that data augmentation is a necessary component of any Computational Pathology application in order to obtain models that are robust to stain variation. Additionally, we provided a ranking of the different techniques evaluated in our experiments, and a protocol for future use.

In **Chapter 4**, we introduced the problem of whole-slide image classification. We developed a novel methodology to train a model that can perform image classification of an entire gigapixel histopathology slide. This method, known as Neural Image Compression, consists of two parts. First, an unsupervisedly trained patch encoder compresses all patches contained in a whole-slide image at very high magnification, resulting in a substantial reduction in the spatial dimensions. Second, a convolutional neural network classifier is trained on these compressed volumes, targeting patient-level labels, without the need for pixel-level annotations. We demonstrated that our method is effective in both natural and histopathology images, resulting in a publication featured in the prestigious IEEE Transactions on Pattern Analysis and Machine Intelligence journal.

In **Chapter 5**, we extended the idea of Neural Image Compression by focusing on improving the encoding mechanism. Instead of using unsupervised training to build the encoder network, we proposed to use supervised multitask training. We found that this procedure resulted in an encoder that can extract highly discriminative and transferable features from any kind of tissue. This improvement in performance allowed us to apply the method to two challenging real-world histopathology datasets. First, we obtained state-of-the-art results in predicting tumor proliferation speed in breast cancer. Second, we were able to train the model using patient overall survival as the reference standard, and obtained satisfactory results when predicting the chance of survival for patients with colorectal metastasis in the liver. These positive results support Neural Image Compression as one of the most promising solutions for the problem of whole-slide image classification.

In **Chapter 6**, we reflected on the main findings and contributions of this thesis. We analyzed the advances and impact in the field, as well as the existing limitations of the proposed methods. Additionally, we provided a future outlook for research opportunities in the field of Computational Pathology with deep learning.

Samenvatting

In **hoofdstuk 1** introduceerden wij de ideeën die ten grondslag lagen aan dit proefschrift en gaven wij een uitgebreide achtergrond. Specifiek ging dit hoofdstuk dieper in op het veld van de *computationele pathologie*, en hoe dit zich verhoudt tot de histopathologie en kunstmatige intelligentie. Histopathologie is de studie van microscopische structuren in menselijk weefsel met als doel het begrijpen van ziekteprocessen. Waar dit veld tot voor kort werd gedomineerd door het gebruik van microscopen, maakt het door de opkomst van nieuwe technologieën een digitale transformatie door. Deze nieuwe methoden voor beeldanalyse heeft pathologen de mogelijkheid gegeven om steeds meer gebruik te maken van algoritmen, zowel binnen het onderzoek als de klinische diagnostiek. Specifiek algoritmes gebaseerd op *deep learning* domineren sinds enkele jaren het veld van de beeldanalyse door hun bewezen effectiviteit. In dit eerste hoofdstuk beschreven we de basis om de revolutie van computationele pathologie te begrijpen en introduceerden we de beoogde doelen van dit proefschrift:

- Hoe om te gaan met fundamentele uitdagingen binnen de computationele pathologie, waaronder: het genereren van annotaties op gigapixel-schaal, variaties in kleuringen tussen laboratoria, en het in één keer kunnen analyseren van gedigitaliseerde weefselcoupes (*whole-slide images*);

- Het automatiseren van een cruciaal component van borstkankergradering: het herkennen van kerndelingen (mitose), en dit inzichtelijk maken voor pathologen;

- Het ontwerpen van een innovatieve methode die classificatie kan uitvoeren op gigapixel whole-slide afbeeldingen op basis van patiënt overleving.

In **hoofdstuk 2** introduceerden wij een methode om automatisch mitose te detecteren in H&E-gekleurde gescande weefselcoupes van borstkankerpatiënten. De toevoeging van deze methode aan het veld kan samengevat worden in drie onderdelen. Ten eerste combineerde onze methode beelden gekleurd met HE en immuunhistochemie. Deze methode resulteerde in de, tot op heden, grootste verzamelde dataset voor mitosedetectie. Als tweede ontwikkelden wij een procedure die ervoor zorgde dat de mitosedetectie robuust was tegen variaties in de kleuring van het weefsel. Als derde zorgde ons trainingsprotocol voor een maximale prestatie, mede door het includeren van additionele moeilijke negatieve voorbeelden, het samenvoegen van meerdere modellen en het comprimeren van het model door middel van *knowledge distillation*. Het resulterende mitose-algoritme is toegepast op beelden van duizenden patiënten uit tientallen centra rond de wereld, gebruikt in verschillende andere

artikelen, en heeft zich bewezen als een belangrijke onderzoekstool binnen het medische domein.

In **hoofdstuk 3** focusten wij op het probleem van kleuringsvariatie tussen én binnen dezelfde laboratoria, en het effect hiervan op de prestaties van neurale netwerken voor enkele applicaties binnen de computationele pathologie. We analyseerden twee mogelijke oplossingen voor dit probleem: het artificieel variëren van de kleuring tijdens trainen, en normalisatie van de kleuring vooraf. Voor beide aanpakken vergeleken we systematisch enkele van de meest gebruikte technieken en introduceerden we onze eigen nieuwe methodes. Na het uitvoeren van honderden experimenten konden we concluderen dat het artificieel variëren van trainingsdata cruciaal is voor elke applicatie binnen de computationele pathologie, met name om robuust te zijn tegen variaties in de kleuring. Als laatste rangschikten we de verschillende technieken, zoals geëvalueerd in onze experimenten, en stelden we een protocol op voor toekomstig gebruik.

In **hoofdstuk 4** introduceerden we het probleem van het classificeren van gedigitaliseerde weefselcoupes. We ontwikkelden een nieuwe methode om een model te trainen dat in één keer een volledige histopathologische weefselcoupe kan classificeren, zonder deze eerst op te splitsen in kleinere afbeeldingen. Deze methode, *Neural Image Compression* genoemd, bestaat uit twee stappen. Eerst wordt er zonder supervisie een encoder getraind die kleine delen van het weefselbeeld (*patches* genoemd) op een hoge vergroting kan comprimeren en op die manier de afmetingen van de gehele afbeelding kan reduceren. Als tweede stap wordt er een neuraal netwerk getraind op deze gecomprimeerde volumes, met als doel het voorspellen van uitkomsten op patiëntniveau. Voor de toepassing van deze methode zijn geen gedetailleerde annotaties nodig op het weefselbeeld. We lieten zien dat onze methode van toepassing is op zowel natuurlijke beelden als gescande weefselcoupes binnen de histopathologie. Onze resultaten werden gepubliceerd in het prestigieuze vakblad *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

In **hoofdstuk 5** breidden we het idee van Neural Image Compression uit door te focussen op het verbeteren van de eerste component: het compressie/encoder mechanisme. In plaats van dit netwerk te trainen zonder annotaties, gebruikten we nu gelabelde data vanuit meerdere taken. We vonden dat deze methode resulteerde in een encoder die betekenisvolle data kon extraheren vanuit verschillende weefseltypes. Deze verbetering zorgde ervoor dat we de methode konden toepassen op twee uitdagende datasets in de histopathologie. Als eerste lieten we zien dat we *state-of-*

the-art resultaten konden behalen in het voorspellen van tumorgroei in borstkanker. Ten tweede lieten we zien dat we dit model konden trainen als een voorspeller voor de overleving van patiënten, met een demonstratie op het gebied van colorectale levermetastasen. Beide resultaten laten zien dat Neural Image Compression een veelbelovende techniek is voor het trainen op gehele weefselcoupes.

In **hoofdstuk 6** reflecteerden we op de resultaten en bijdragen van dit proefschrift. We analyseerden de geboekte vooruitgang en impact op het veld, en de beperkingen die gebonden zijn aan de voorgestelde methoden. Afsluitend lieten we enkele toekomstige richtingen zien voor onderzoeksmogelijkheden binnen de computationele pathologie en deep learning.

Publications

Papers in international journals

Tellez D, Litjens G, van der Laak J, and Ciompi F. *Neural image compression for gigapixel histopathology image analysis*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019.

Tellez D, Litjens G, Bándi P, Bulten W, Bokhorst JM, Ciompi F, and van der Laak J. *Quantifying the effects of data augmentation and stain color normalization in convolutional neural networks for computational pathology*. Medical image analysis, 2019.

Tellez D, Balkenhol M, Otte-Höller I, van de Loo R, Vogels R, Bult P, Wauters C, Vreuls W, Mol S, Karssemeijer N, Litjens G, van der Laak J, and Ciompi F. *Whole-Slide Mitosis Detection in H&E Breast Histology Using PHH3 as a Reference to Train Distilled Stain-Invariant Convolutional Networks*. IEEE Transactions on Medical Imaging, 2018.

Balkenhol M, Bult P, **Tellez D**, Vreuls W, Clahsen PC, Ciompi F, and van der Laak J. *Deep learning and manual assessment show that the absolute mitotic count does not contain prognostic information in triple negative breast cancer*. Cellular Oncology, 2019.

Balkenhol M, **Tellez D**, Vreuls W, Clahsen PC, Pinckaers H, Ciompi F, Bult P, and van der Laak J. *Deep learning assisted mitotic counting for breast cancer*. Laboratory Investigation, 2019.

Veta M, Heng YJ, Stathonikos N, Ehteshami Bejnordi B, Beca F, Wollmann T, Rohr K, Shah MA, Wang D, Rousson M, Hedlund M, **Tellez D**, Ciompi F, Zerhouni E, Lanyi D, Viana M, Kovalev V, Liauchuk V, Phoulady HA, Qaiser T, Graham S, Rajpoot N, Sjöblom E, Molin J, Paeng K, Hwang S, Park S, Jia Z, Eric I, Chang C, Xu Y, Beck AH, van Diest P, and Pluim J. *Predicting breast tumor proliferation from whole-slide images: the TUPAC16 challenge*. Medical image analysis, 2019.

Ehteshami Bejnordi B, Veta M, Van Diest P, van Ginneken B, Karssemeijer N, Litjens G, van Der Laak J, Hermsen M, Manson Q, Balkenhol M, Geessink O, Stathonikos N, van Dijk M, Bult P, Beca F, Beck A, Wang D, Khosla A, Gargeya R, Irshad H, Zhong A, Dou Q, Li Q, Chen H, Lin H, Heng PA, Haß C, Bruni E, Wong Q, Halici U, Ümit Öner M, Cetin-Atalay R, Berseth M, Khvatkov V, Vylegzhanin A, Kraus O, Shaban M, Rajpoot N, Awan R, Sirinukunwattana K, Qaiser T, Tsang Y, **Tellez D**, Annuscheit J, Hufnagl P, Valkonen M, Kartasalo K, Latonen L, Ruusuvuori P, Liimatainen K, Albarqouni S, Mungal B, George A, Demirci S, Navab N, Watanabe S, et al. *Diagnostic*

assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. Journal of the American Medical Association, 2017.

Papers in conference proceedings

Tellez D, Hoppener D, Verhoef C, Grunhagen D, Nierop P, Drozdzal M, van der Laak J, and Ciompi F. *Extending Unsupervised Neural Image Compression With Supervised Multitask Learning*. Oral presentation in Medical Imaging with Deep Learning, 2020.

Mercan C, Reijnen-Mooij G, **Tellez D**, Lotz J, Weiss N, van Gerven M, and Ciompi F. *Virtual staining for mitosis detection in Breast Histopathology*. Poster presentation in the IEEE International Symposium on Biomedical Imaging, 2020.

Balkenhol M, Bult P, **Tellez D**, Vreuls W, Clahsen PC, Ciompi F, and van der Laak J. *Artificial intelligence in breast cancer histopathology*. Oral presentation in Belgian Society of Senology Conference, 2020.

Tellez D, van der Laak J, and Ciompi F. *Gigapixel Whole-Slide Image Classification Using Unsupervised Image Compression And Contrastive Training*. Poster presentation in Medical Imaging with Deep Learning, 2018.

Tellez D, Balkenhol M, Karssemeijer N, Litjens G, van der Laak J, and Ciompi F. *H&E stain augmentation improves generalization of convolutional networks for histopathological mitosis detection*. Poster presentation in Medical Imaging of Proceedings of the SPIE, 2018.

Awards

Runner-up for Best Paper Award (third place) in the Medical Imaging with Deep Learning conference, 2020.

TUPAC Grand Challenge: obtained second place in the proliferation prediction task. The International Conference on Medical Image Computing and Computer Assisted Intervention, 2016.

Camelyon Grand Challenge: obtained second place in the lesion-based detection task. The IEEE International Symposium on Biomedical Imaging, 2016.

Media

Best of MIDL20: Extending Unsupervised Neural Image Compression With Supervised Multitask Learning. Computer Vision News, 2020[183].

Using multitask learning to improve image classification for histopathology. Blog of Facebook AI Research, 2020[182].

Bibliography

[1] Kumar V. *Robbins basic pathology*. Elsevier, 2017. ISBN 9780323480543.

[2] Ramos-Vara J. and Miller M. When tissue antigens and antibodies get along: revisiting the technical aspects of immunohistochemistry—the red, brown, and blue technique. *Veterinary pathology*, 51(1):42–87, 2014.

[3] Leader M., Patel J., Makin C., and Henry K. An analysis of the sensitivity and specificity of the cytokeratin marker cam 5.2 for epithelial tumours. results of a study of 203 sarcomas, 50 carcinomas and 28 malignant melanomas. *Histopathology*, 10(12):1315–1324, 1986.

[4] Scholzen T. and Gerdes J. The ki-67 protein: from the known and the unknown. *Journal of cellular physiology*, 182(3):311–322, 2000.

[5] McGuire A., Brown J. A., Malone C., McLaughlin R., and Kerin M. J. Effects of age on the detection and management of breast cancer. *Cancers*, 7(2):908–929, 2015.

[6] Balasubramanian R., Rolph R., Morgan C., and Hamed H. Genetics of breast cancer: management strategies and risk-reducing surgery. *British Journal of Hospital Medicine*, 80(12):720–725, 2019.

[7] Saslow D., Hannan J., Osuch J., Alciati M. H., Baines C., Barton M., Bobo J. K., Coleman C., Dolan M., Gaumer G., et al. Clinical breast examination: practical recommendations for optimizing performance and reporting. *CA: a cancer journal for clinicians*, 54(6):327–344, 2004.

[8] Qaseem A., Lin J. S., Mustafa R. A., Horwitch C. A., and Wilt T. J. Screening for breast cancer in average-risk women: a guidance statement from the american college of physicians. *Annals of internal medicine*, 170(8):547–560, 2019.

[9] Sinn H.-P. and Kreipe H. A brief overview of the WHO classification of breast tumors. *Breast care*, 8(2):149–154, 2013.

[10] Elston C. W. and Ellis I. O. Pathological prognostic factors in breast cancer. i. the value of histological grade in breast cancer: experience from a large study with long-term follow-up. *Histopathology*, 19(5):403–410, 1991.

[11] Elston C. W., Ellis I. O., and Pinder S. E. Pathological prognostic factors in breast cancer. *Critical reviews in oncology/hematology*, 31(3):209–223, 1999.

[12] Skaland I., van Diest P. J., Janssen E. A., Gudlaugsson E., and Baak J. P. Prognostic differences of world health organization–assessed mitotic activity index and mitotic impression by quick scanning in invasive ductal breast cancer patients younger than 55 years. *Human pathology*, 39 (4):584–590, 2008.

[13] Dent R., Trudeau M., Pritchard K. I., Hanna W. M., Kahn H. K., Sawka C. A., Lickley L. A., Rawlinson E., Sun P., and Narod S. A. Triple-negative breast cancer: clinical features and patterns of recurrence. *Clinical cancer research*, 13(15):4429–4434, 2007.

[14] Abels E., Pantanowitz L., Aeffner F., Zarella M. D., van der Laak J., Bui M. M., Vemuri V. N., Parwani A. V., Gibbs J., Agosto-Arroyo E., et al. Computational pathology definitions, best practices, and recommendations for regulatory guidance: a white paper from the digital pathology association. *The Journal of pathology*, 249(3):286–294, 2019.

[15] Litjens G. Automated slide analysis platform (ASAP), 2017.

[16] Dimitriou N., Arandjelović O., and Caie P. D. Deep learning for whole slide image analysis: An

overview. *Frontiers in Medicine*, 6, 2019.

[17] Russell S. and Norvig P. Artificial intelligence: a modern approach. 2002.

[18] Goodfellow I., Bengio Y., and Courville A. *Deep Learning*. MIT Press, 2016.

[19] LeCun Y., Bengio Y., and Hinton G. Deep learning. *nature*, 521(7553):436–444, 2015.

[20] Krizhevsky A. et al. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[21] Guo Y., Liu Y., Oerlemans A., Lao S., Wu S., and Lew M. S. Deep learning for visual understanding: A review. *Neurocomputing*, 187:27–48, 2016.

[22] Purwins H., Li B., Virtanen T., Schlüter J., Chang S.-Y., and Sainath T. Deep learning for audio signal processing. *IEEE Journal of Selected Topics in Signal Processing*, 13(2):206–219, 2019.

[23] Nassif A. B., Shahin I., Attili I., Azzeh M., and Shaalan K. Speech recognition using deep neural networks: A systematic review. *IEEE Access*, 7:19143–19165, 2019.

[24] Brown T. B., Mann B., Ryder N., Subbiah M., Kaplan J., Dhariwal P., Neelakantan A., Shyam P., Sastry G., Askell A., et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[25] Wu Z., Pan S., Chen F., Long G., Zhang C., and Philip S. Y. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[26] Nguyen T. T., Nguyen N. D., and Nahavandi S. Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications. *IEEE transactions on cybernetics*, 2020.

[27] Pan Z., Yu W., Yi X., Khan A., Yuan F., and Zheng Y. Recent progress on generative adversarial networks (gans): A survey. *IEEE Access*, 7:36322–36333, 2019.

[28] Litjens G., Kooi T., Bejnordi B. E., Setio A. A. A., Ciompi F., Ghafoorian M., van der Laak J. A., van Ginneken B., and Sánchez C. I. A survey on deep learning in medical image analysis. *Medical Image Analysis*, 42:60 – 88, 2017.

[29] Freund Y. and Schapire R. E. Large margin classification using the perceptron algorithm. *Machine learning*, 37(3):277–296, 1999.

[30] Kline D. M. and Berardi V. L. Revisiting squared-error and cross-entropy functions for training neural network classifiers. *Neural Computing & Applications*, 14(4):310–318, 2005.

[31] Bottou L. and Bousquet O. The tradeoffs of large scale learning. In *Advances in Neural Information Processing Systems 20 (NIPS 2007)*, pages 161–168. 2008.

[32] Kolen J. F. and Kremer S. C. Gradient flow in recurrent nets: The difficulty of learning longterm dependencies. 2001.

[33] Glorot X., Bordes A., and Bengio Y. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323, 2011.

[34] He K., Zhang X., Ren S., and Sun J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[35] Kim D., Kim J., and Kim J. Elastic exponential linear units for convolutional neural networks.

Neurocomputing, 2020.

[36] He K., Zhang X., Ren S., and Sun J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[37] Qian N. On the momentum term in gradient descent learning algorithms. *Neural networks*, 12 (1):145–151, 1999.

[38] Tieleman T. and Hinton G. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31, 2012.

[39] Kingma D. P. and Ba J. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2014.

[40] Ioffe S. and Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.

[41] LeCun Y., Boser B., Denker J. S., Henderson D., Howard R. E., Hubbard W., and Jackel L. D. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[42] Zeiler M. D. and Fergus R. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[43] Dumoulin V. and Visin F. A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*, 2016.

[44] Radford A., Metz L., and Chintala S. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[45] Wei Y., Xiao H., Shi H., Jie Z., Feng J., and Huang T. S. Revisiting dilated convolution: A simple approach for weakly-and semi-supervised semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7268–7277, 2018.

[46] Chollet F. Xception: Deep learning with depthwise separable convolutions. *arXiv preprint*, 2016.

[47] Howard A. G., Zhu M., Chen B., Kalenichenko D., Wang W., Weyand T., Andreetto M., and Adam H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[48] Srivastava N., Hinton G., Krizhevsky A., Sutskever I., and Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1): 1929–1958, 2014.

[49] Bloom H. and Richardson W. Histological grading and prognosis in breast cancer: a study of 1409 cases of which 359 have been followed for 15 years. *British journal of cancer*, 11(3):359, 1957.

[50] Van Diest P., Van Der Wall E., and Baak J. Prognostic value of proliferation in invasive breast cancer: a review. *Journal of clinical pathology*, 57(7):675–681, 2004.

[51] Al-Janabi S., van Slooten H.-J., Visser M., Van Der Ploeg T., van Diest P. J., and Jiwa M. Evaluation of mitotic activity index in breast cancer using whole slide digital images. *PloS one*, 8(12): e82576, 2013.

[52] Roux L., Racoceanu D., Loménie N., Kulikova M., Irshad H., Klossa J., Capron F., Genestie C., Le Naour G., Gurcan M. N., et al. Mitosis detection in breast cancer histological images an icpr

2012 contest. *Journal of pathology informatics*, 4(1):8, 2013.

[53] Veta M., Van Diest P. J., Willems S. M., Wang H., Madabhushi A., Cruz-Roa A., Gonzalez F., Larsen A. B., Vestergaard J. S., Dahl A. B., et al. Assessment of algorithms for mitosis detection in breast cancer histopathology images. *Medical image analysis*, 20(1):237–248, 2015.

[54] Roux L. Mitosis detection in breast cancer histological images, 2014. `https://mitos-atypia-14.grand-challenge.org`.

[55] Veta M. Tumor proliferation assessment challenge, 2016. `http://tupac.tue-image.nl`.

[56] Krizhevsky A. and Hinton G. Learning multiple layers of features from tiny images. 2009.

[57] Deng J., Dong W., Socher R., Li L.-J., Li K., and Fei-Fei L. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[58] Cireşan D. C., Giusti A., Gambardella L. M., and Schmidhuber J. Mitosis detection in breast cancer histology images with deep neural networks. In *International Conference on Medical Image Computing and Computer-assisted Intervention*, pages 411–418. Springer, 2013.

[59] Veta M., van Diest P. J., Jiwa M., Al-Janabi S., and Pluim J. P. Mitosis counting in breast cancer: Object-level interobserver agreement and comparison to an automatic method. *PloS one*, 11(8): e0161286, 2016.

[60] Ribalta T., McCutcheon I. E., Aldape K. D., Bruner J. M., and Fuller G. N. The mitosis-specific antibody anti-phosphohistone-h3 (phh3) facilitates rapid reliable grading of meningiomas according to who 2000 criteria. *The American journal of surgical pathology*, 28(11):1532–1536, 2004.

[61] Focke C. M., Finsterbusch K., Decker T., and van Diest P. J. Performance of 4 immunohistochemical phosphohistone h3 antibodies for marking mitotic figures in breast cancer. *Applied Immunohistochemistry & Molecular Morphology*, 26(1):20–26, 2018.

[62] Skaland I., Janssen E. A., Gudlaugsson E., Klos J., Kjellevold K. H., Søiland H., and Baak J. P. Validating the prognostic value of proliferation measured by phosphohistone h3 (pph3) in invasive lymph node-negative breast cancer patients less than 71 years of age. *Breast cancer research and treatment*, 114(1):39–45, 2009.

[63] Colman H., Giannini C., Huang L., Gonzalez J., Hess K., Bruner J., Fuller G., Langford L., Pelloski C., Aaron J., et al. Assessment and prognostic significance of mitotic index using the mitosis marker phospho-histone h3 in low and intermediate-grade infiltrating astrocytomas. *The American journal of surgical pathology*, 30(5):657–664, 2006.

[64] Fukushima S., Terasaki M., Sakata K., Miyagi N., Kato S., Sugita Y., and Shigemori M. Sensitivity and usefulness of anti-phosphohistone-h3 antibody immunostaining for counting mitotic figures in meningioma cases. *Brain tumor pathology*, 26(2):51–57, 2009.

[65] Ikenberg K., Pfaltz M., Rakozy C., and Kempf W. Immunohistochemical dual staining as an adjunct in assessment of mitotic activity in melanoma. *Journal of cutaneous pathology*, 39(3): 324–330, 2012.

[66] Zerhouni E., Lányi D., Viana M., and Gabrani M. Wide residual networks for mitosis detection. In *Biomedical Imaging (ISBI 2017), 2017 IEEE 14th International Symposium on*, pages 924–928. IEEE, 2017.

[67] Paeng K., Hwang S., Park S., Kim M., and Kim S. A unified framework for tumor proliferation score prediction in breast histopathology. *arXiv preprint arXiv:1612.07180*, 2016.

[68] Macenko M., Niethammer M., Marron J., Borland D., Woosley J. T., Guan X., Schmitt C., and Thomas N. E. A method for normalizing histology slides for quantitative analysis. In *Biomedical Imaging: From Nano to Macro, 2009. ISBI'09. IEEE International Symposium on*, pages 1107–1110. IEEE, 2009.

[69] Khan A. M., Rajpoot N., Treanor D., and Magee D. A nonlinear mapping approach to stain normalization in digital histopathology images using image-specific color deconvolution. *IEEE Transactions on Biomedical Engineering*, 61(6):1729–1738, 2014.

[70] Bejnordi B. E., Litjens G., Timofeeva N., Otte-Höller I., Homeyer A., Karssemeijer N., and van der Laak J. A. Stain specific standardization of whole-slide histopathological images. *IEEE transactions on medical imaging*, 35(2):404–415, 2016.

[71] Liu Y., Gadepalli K., Norouzi M., Dahl G. E., Kohlberger T., Boyko A., Venugopalan S., Timofeev A., Nelson P. Q., Corrado G. S., et al. Detecting cancer metastases on gigapixel pathology images. *arXiv preprint arXiv:1703.02442*, 2017.

[72] Hinton G., Vinyals O., and Dean J. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[73] Weinstein J. N., Collisson E. A., Mills G. B., Shaw K. M., Ozenberger B. A., Ellrott K., Shmulevich I., Sander C., Stuart J. M., Network C. G. A. R., et al. The cancer genome atlas pan-cancer analysis project. *Nature genetics*, 45(10):1113, 2013.

[74] Nielsen T. O., Parker J. S., Leung S., Voduc D., Ebbert M., Vickery T., Davies S. R., Snider J., Stijleman I. J., Reed J., et al. A comparison of pam50 intrinsic subtyping with immunohistochemistry and clinical prognostic factors in tamoxifen-treated estrogen receptor–positive breast cancer. *Clinical cancer research*, pages 1078–0432, 2010.

[75] Bándi P., van de Loo R., Intezar M., Geijs D., Ciompi F., van Ginneken B., van der Laak J., and Litjens G. Comparison of different methods for tissue segmentation in histopathological whole-slide images. *arXiv preprint arXiv:1703.05990*, 2017.

[76] Simard P. Y., Steinkraus D., Platt J. C., et al. Best practices for convolutional neural networks applied to visual document analysis. In *ICDAR*, volume 3, pages 958–962. Citeseer, 2003.

[77] Ruifrok A. C., Johnston D. A., et al. Quantification of histochemical staining by color deconvolution. *Analytical and quantitative cytology and histology*, 23(4):291–299, 2001.

[78] Springenberg J. T., Dosovitskiy A., Brox T., and Riedmiller M. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

[79] Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688, May 2016.

[80] Dieleman S., Schlüter J., Raffel C., Olson E., Sønderby S. K., Nouri D., et al. Lasagne: First release., August 2015. URL `http://dx.doi.org/10.5281/zenodo.27878`.

[81] Komura D. and Ishikawa S. Machine learning methods for histopathological image analysis. *Computational and structural biotechnology journal*, 16:34–42, 2018.

[82] Veta M., Heng Y. J., Stathonikos N., Bejnordi B. E., Beca F., Wollmann T., Rohr K., Shah M. A.,

Wang D., Rousson M., et al. Predicting breast tumor proliferation from whole-slide images: the TUPAC16 challenge. *Medical Image Analysis*, 54:111–121, 2019.

[83] Sirinukunwattana K. et al. Gland segmentation in colon histology images: The glas challenge contest. *Medical image analysis*, 35:489–502, 2017.

[84] Tellez D., Balkenhol M., Otte-Höller I., van de Loo R., Vogels R., Bult P., Wauters C., Vreuls W., Mol S., Karssemeijer N., et al. Whole-Slide Mitosis Detection in H&E Breast Histology Using PHH3 as a Reference to Train Distilled Stain-Invariant Convolutional Networks. *IEEE Transactions on Medical Imaging*, 37(9):2126–2136, 2018.

[85] Bug D., Schneider S., Grote A., Oswald E., Feuerhake F., Schüler J., and Merhof D. Context-based normalization of histological stains using deep convolutional features. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pages 135–142. Springer, 2017.

[86] Reinhard E., Adhikhmin M., Gooch B., and Shirley P. Color transfer between images. *IEEE Computer Graphics and Applications*, 21(5):34–41, 2001.

[87] Kingma D. P. and Welling M. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2013.

[88] Goodfellow I., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A., and Bengio Y. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

[89] Cho H., Lim S., Choi G., and Min H. Neural stain-style transfer learning using GAN for histopathological images. In *Asian Conference on Machine Learning*, 2017.

[90] Zanjani F. G., Zinger S., Bejnordi B. E., van der Laak J. A., and de With P. H. Stain normalization of histopathology images using generative adversarial networks. In *International Symposium on Biomedical Imaging*, pages 573–577. IEEE, 2018.

[91] Clarke E. L. and Treanor D. Colour in digital pathology: a review. *Histopathology*, 70(2):153–163, 2017.

[92] Albarqouni S., Baur C., Achilles F., Belagiannis V., Demirci S., and Navab N. Aggnet: deep learning from crowds for mitosis detection in breast cancer histology images. *IEEE Transactions on Medical Imaging*, 35(5):1313–1321, 2016.

[93] Janowczyk A., Basavanhally A., and Madabhushi A. Stain normalization using sparse autoencoders (stanosa): application to digital pathology. *Computerized Medical Imaging and Graphics*, 57:50–61, 2017.

[94] Bándi P., Geessink O., Manson Q., van Dijk M., Balkenhol M., Hermsen M., Bejnordi B. E., Lee B., Paeng K., Zhong A., et al. From detection of individual metastases to classification of lymph node status at the patient level: the CAMELYON17 challenge. *IEEE Transactions on Medical Imaging*, 38(2):550–560, 2019.

[95] Wang D., Foran D. J., Ren J., Zhong H., Kim I. Y., and Qi X. Exploring automatic prostate histopathology image gleason grading via local structure modeling. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 2649–2652. IEEE, 2015.

[96] Zhu Y., Zhang S., Liu W., and Metaxas D. N. Scalable histopathological image analysis via active learning. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 369–376. Springer, 2014.

[97] Bulten W., Bándi P., Hoven J., van de Loo R., Lotz J., Weiss N., van der Laak J., van Ginneken B., Hulsbergen-van de Kaa C., and Litjens G. Epithelium segmentation using deep learning in H&E-stained prostate specimens with immunohistochemistry as reference standard. *Scientific Reports*, 9(1):864, 2019.

[98] Ciompi F., Geessink O., Bejnordi B. E., de Souza G. S., Baidoshvili A., Litjens G., van Ginneken B., Nagtegaal I., and van der Laak J. The importance of stain normalization in colorectal tissue classification with convolutional networks. In *Biomedical Imaging (ISBI 2017), 2017 IEEE 14th International Symposium on*, pages 160–163. IEEE, 2017.

[99] Gertych A., Ing N., Ma Z., Fuchs T. J., Salman S., Mohanty S., Bhele S., Velásquez-Vacca A., Amin M. B., and Knudsen B. S. Machine learning approaches to analyze histological images of tissues from radical prostatectomies. *Computerized Medical Imaging and Graphics*, 46:197–208, 2015.

[100] Kather J. N., Weis C.-A., Bianconi F., Melchers S. M., Schad L. R., Gaiser T., Marx A., and Zöllner F. G. Multi-class texture analysis in colorectal cancer histology. *Scientific Reports*, 6:27988, 2016.

[101] Haeberli P. and Voorhies D. Image processing by linear interpolation and extrapolation. *IRIS Universe Magazine*, 28:8–9, 1994.

[102] Van der Walt S., Schönberger J. L., Nunez-Iglesias J., Boulogne F., Warner J. D., Yager N., Gouillart E., and Yu T. Scikit-image: image processing in python. *PeerJ*, 2, 2014.

[103] Bejnordi B. E., Veta M., van Diest P. J., van Ginneken B., Karssemeijer N., Litjens G., van der Laak J. A., Hermsen M., Manson Q. F., Balkenhol M., et al. Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. *Jama*, 318(22):2199–2210, 2017.

[104] Ronneberger O., Fischer P., and Brox T. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[105] Maas A. L., Hannun A. Y., and Ng A. Y. Rectifier nonlinearities improve neural network acoustic models. In *International Conference on Machine Learning*, 2013.

[106] Odena A., Dumoulin V., and Olah C. Deconvolution and checkerboard artifacts. *Distill*, 1(10), 2016.

[107] Demšar J. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7(Jan):1–30, 2006.

[108] Zhang L. et al. Deep learning for remote sensing data: A technical tutorial on the state of the art. *IEEE Geoscience and Remote Sensing Magazine*, 4(2):22–40, 2016.

[109] Ehteshami Bejnordi B. et al. Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. *JAMA*, 2017.

[110] Wang X. et al. Weakly supervised learning for whole slide lung cancer image classification. In *Medical Imaging with Deep Learning*, 2018.

[111] Ilse M. et al. Attention-based deep multiple instance learning. *arXiv preprint arXiv:1802.04712*, 2018.

[112] Combalia M. et al. Monte-carlo sampling applied to multiple instance learning for whole slide image classification. In *Medical Imaging with Deep Learning*, 2018.

[113] Tomczak J. M. et al. Histopathological classification of precursor lesions of esophageal adenocarcinoma: A deep multiple instance learning approach. In *Medical Imaging with Deep Learning*, 2018.

[114] Hou L. et al. Patch-based convolutional neural network for whole slide tissue image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[115] Coudray N. et al. Classification and mutation prediction from non-small cell lung cancer histopathology images using deep learning. *Nature medicine*, 2018.

[116] Theis L. et al. Lossy image compression with compressive autoencoders. *arXiv preprint arXiv:1703.00395*, 2017.

[117] Donahue J. et al. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016.

[118] Dumoulin V. et al. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2016.

[119] Oord A. v. d. et al. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[120] van den Oord A. et al. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, pages 6306–6315, 2017.

[121] Koch G. et al. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, volume 2, 2015.

[122] Melekhov I. et al. Siamese network features for image matching. In *Pattern Recognition (ICPR), 2016 23rd International Conference on*, pages 378–383. IEEE, 2016.

[123] Hyvarinen A. et al. Unsupervised feature extraction by time-contrastive learning and nonlinear ica. In *Advances in Neural Information Processing Systems*, pages 3765–3773, 2016.

[124] Goodfellow I. et al. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27*, pages 2672–2680. 2014.

[125] Chen X. et al. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, 2016.

[126] Selvaraju R. R. et al. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, pages 618–626, 2017.

[127] D. Tellez et al. Gigapixel Whole-Slide Image Classification Using Unsupervised Image Compression And Contrastive Training. In *Medical Imaging with Deep Learning*, 2018.

[128] LeCun Y. et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[129] Sertel O. et al. Computer-aided prognosis of neuroblastoma on whole-slide images: Classification of stromal development. *Pattern recognition*, 42(6):1093–1103, 2009.

[130] Tabesh A. et al. Multifeature prostate cancer diagnosis and gleason grading of histological images. *IEEE Transactions on Medical Imaging*, 26(10):1366–1378, 2007.

A

[131] Kong J. et al. Image analysis for automated assessment of grade of neuroblastic differentiation. In *2007 4th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 61–64. IEEE, 2007.

[132] Shin H.-C. et al. Learning to read chest x-rays: Recurrent neural cascade model for automated image annotation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[133] Hasan S. A. et al. Overview of the ImageCLEF 2018 medical domain visual question answering task. In *CLEF2018 Working Notes*, CEUR Workshop Proceedings, 2018.

[134] Anavi Y. et al. A comparative study for chest radiograph image retrieval using binary texture and deep learning classification. In *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*, 2015.

[135] Schlegl T. et al. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *International Conference on Information Processing in Medical Imaging*, 2017.

[136] Yang Q. et al. Low dose ct image denoising using a generative adversarial network with wasserstein distance and perceptual loss. *IEEE transactions on medical imaging*, 2018.

[137] Bang D. et al. High quality bidirectional generative adversarial networks. *arXiv preprint arXiv:1805.10717*, 2018.

[138] Caron M. et al. Deep clustering for unsupervised learning of visual features. *arXiv preprint arXiv:1807.05520*, 2018.

[139] Jetley S. et al. Learn to pay attention. *arXiv preprint arXiv:1804.02391*, 2018.

[140] Chen T. et al. Training deep nets with sublinear memory cost. *arXiv preprint arXiv:1604.06174*, 2016.

[141] Chollet F. et al. Keras. `https://keras.io`, 2015.

[142] Pawlowski N., Bhooshan S., Ballas N., Ciompi F., Glocker B., and Drozdzal M. Needles in Haystacks: On Classifying Tiny Objects in Large Images. *arXiv preprint*, 2019.

[143] Xu Y., Jia Z., Wang L.-B., Ai Y., Zhang F., Lai M., Eric I., and Chang C. Large scale tissue histopathology image classification, segmentation, and visualization via deep convolutional activation features. *BMC bioinformatics*, 18(1):281, 2017.

[144] Quellec G., Cazuguel G., Cochener B., and Lamard M. Multiple-instance learning for medical image and video analysis. *IEEE reviews in biomedical engineering*, 10:213–234, 2017.

[145] Qaiser T. and Rajpoot N. M. Learning where to see: A novel attention model for automated immunohistochemical scoring. *IEEE transactions on medical imaging*, 2019.

[146] Dong N., Kampffmeyer M., Liang X., Wang Z., Dai W., and Xing E. Reinforced auto-zoom net: Towards accurate and fast breast cancer segmentation in whole-slide images. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pages 317–325. Springer, 2018.

[147] BenTaieb A. and Hamarneh G. Predicting cancer with a recurrent visual attention model for histopathology images. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 129–137. Springer, 2018.

[148] Pinckaers H., van Ginneken B., and Litjens G. Streaming convolutional neural networks for end-to-end learning with multi-megapixel images. *arXiv preprint arXiv:1911.04432*, 2019.

[149] Tellez D., Litjens G., van der Laak J., and Ciompi F. Neural image compression for gigapixel histopathology image analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43 (2):567–578, 2021.

[150] Tan M. and Le Q. EfficientNet: Rethinking model scaling for convolutional neural networks. In Chaudhuri K. and Salakhutdinov R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6105–6114, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

[151] Caruana R. Multitask learning. *Machine Learning*, 28(1):41–75, Jul 1997.

[152] Ruder S. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

[153] Zhang Y. and Yang Q. A survey on multi-task learning. *arXiv preprint arXiv:1707.08114*, 2017.

[154] Galjart B., Nierop P. M., van der Stok E. P., van den Braak R. R. C., Höppener D. J., Daelemans S., Dirix L. Y., Verhoef C., Vermeulen P. B., and Grünhagen D. J. Angiogenic desmoplastic histopathological growth pattern as a prognostic marker of good outcome in patients with colorectal liver metastases. *Angiogenesis*, 22(2):355–368, 2019.

[155] Tellez D., Litjens G., Bándi P., Bulten W., Bokhorst J.-M., Ciompi F., and van der Laak J. Quantifying the effects of data augmentation and stain color normalization in convolutional neural networks for computational pathology. *Medical Image Analysis*, 58:101544, 2019.

[156] Van Dam P.-J., Van Der Stok E. P., Teuwen L.-A., Van den Eynden G. G., Illemann M., Frentzas S., Majeed A. W., Eefsen R. L., Van Den Braak R. R. C., Lazaris A., et al. International consensus guidelines for scoring the histopathological growth patterns of liver metastasis. *British journal of cancer*, 117(10):1427–1441, 2017.

[157] Höppener D., Nierop P., Herpel E., Rahbari N., Doukas M., Vermeulen P., Grünhagen D., and Verhoef C. Histopathological growth patterns of colorectal liver metastasis exhibit little heterogeneity and can be determined with a high diagnostic accuracy. *Clinical & experimental metastasis*, pages 1–9, 2019.

[158] Faraggi D. and Simon R. A neural network model for survival data. *Statistics in medicine*, 14(1): 73–82, 1995.

[159] Kendall A., Gal Y., and Cipolla R. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7482–7491, 2018.

[160] Chen Z., Badrinarayanan V., Lee C.-Y., and Rabinovich A. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International Conference on Machine Learning*, pages 794–803, 2018.

[161] Zamir A. R., Sax A., Shen W. B., Guibas L. J., Malik J., and Savarese S. Taskonomy: Disentangling task transfer learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018.

[162] Cox D. R. Regression models and life-tables. *Journal of the Royal Statistical Society: Series B*

(Methodological), 34(2):187–202, 1972.

[163] Hermsen M., de Bel T., Den Boer M., Steenbergen E. J., Kers J., Florquin S., Roelofs J. J., Stegall M. D., Alexander M. P., Smith B. H., et al. Deep learning–based histopathologic assessment of kidney tissue. *Journal of the American Society of Nephrology*, 30(10):1968–1979, 2019.

[164] Gupta A., Harrison P. J., Wieslander H., Pielawski N., Kartasalo K., Partel G., Solorzano L., Suveer A., Klemm A. H., Spjuth O., et al. Deep learning in image cytometry: a review. *Cytometry Part A*, 95(4):366–380, 2019.

[165] Akram S. U., Qaiser T., Graham S., Kannala J., Heikkilä J., and Rajpoot N. Leveraging unlabeled whole-slide-images for mitosis detection. In *Computational Pathology and Ophthalmic Medical Image Analysis*, pages 69–77. Springer, 2018.

[166] Sing T., Hoefling H., Hossain I., Boisclair J., Doelemeyer A., Flandre T., Piaia A., Romanet V., Santarossa G., Saravanan C., et al. A deep learning-based model of normal histology. *bioRxiv*, page 838417, 2019.

[167] Lampert T., Merveille O., Schmitz J., Forestier G., Feuerhake F., and Wemmert C. Strategies for training stain invariant cnns. In *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, pages 905–909. IEEE, 2019.

[168] Lafarge M., Pluim J., Eppenhof K., and Veta M. Learning domain-invariant representations of histological images. *Frontiers in medicine*, 6:162, 2019.

[169] Otálora S., Atzori M., Andrearczyk V., Khan A., and Müller H. Staining invariant features for improving generalization of deep convolutional neural networks in computational pathology. *Frontiers in Bioengineering and Biotechnology*, 7:198, 2019.

[170] van der Laak J., Ciompi F., and Litjens G. No pixel-level annotations needed. *Nature Biomedical Engineering*, 3(11):855–856, 2019.

[171] Ibrahim A., Gamble P., Jaroensri R., Abdelsamea M. M., Mermel C. H., Chen P.-H. C., and Rakha E. A. Artificial intelligence in digital breast pathology: Techniques and applications. *The Breast*, 2019.

[172] Kumar N., Verma R., Anand D., Zhou Y., Onder O. F., Tsougenis E., Chen H., Heng P. A., Li J., Hu Z., et al. A multi-organ nucleus segmentation challenge. *IEEE transactions on medical imaging*, 2019.

[173] Ianni J. D., Soans R. E., Sankarapandian S., Chamarthi R. V., Ayyagari D., Olsen T. G., Bonham M. J., Stavish C. C., Motaparthi K., Cockerell C. J., et al. Augmenting the pathology lab: An intelligent whole slide image classification system for the real world. *arXiv preprint arXiv:1909.11212*, 2019.

[174] Seth N. *Automated Localization of Breast Ductal Carcinoma in Situ in Whole Slide Images*. PhD thesis, 2019.

[175] Bándi P., Balkenhol M., van Ginneken B., van der Laak J., and Litjens G. Resolution-agnostic tissue segmentation in whole-slide histopathology images with convolutional neural networks. *PeerJ*, 7:e8242, 2019.

[176] Saxena S. and Gyanchandani M. Machine learning methods for computer-aided breast cancer diagnosis using histopathology: A narrative review. *Journal of Medical Imaging and Radiation*

Sciences, 2019.

[177] Arvidsson I., Overgaard N. C., Åström K., and Heyden A. Comparison of different augmentation techniques for improved generalization performance for gleason grading. In *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, pages 923–927. IEEE, 2019.

[178] Hesse L. S., Kuling G., Veta M., and Martel A. L. Intensity augmentation for domain transfer of whole breast segmentation in mri. *arXiv preprint arXiv:1909.02642*, 2019.

[179] Stacke K., Eilertsen G., Unger J., and Lundström C. A closer look at domain shift for deep learning in histopathology. *arXiv preprint arXiv:1909.11575*, 2019.

[180] Balkenhol M. C., Tellez D., Vreuls W., Clahsen P. C., Pinckaers H., Ciompi F., Bult P., and van der Laak J. A. Deep learning assisted mitotic counting for breast cancer. *Laboratory Investigation*, page 1, 2019.

[181] Balkenhol M. C., Bult P., Tellez D., Vreuls W., Clahsen P. C., Ciompi F., and van der Laak J. A. Deep learning and manual assessment show that the absolute mitotic count does not contain prognostic information in triple negative breast cancer. *Cellular Oncology*, pages 1–15, 2019.

[182] Facebook AI Research. Using multitask learning to improve image classification for histopathology, 2020. URL `https://ai.facebook.com/blog/using-multitask-learning-to-improve-image-classification-for-histopathology/`.

[183] Computer Vision News. Extending unsupervised neural image compression with supervised multitask learning, 2020. URL `https://rsipvision.com/ComputerVisionNews-2020August/20/`.

Acknowledgements

I had the privilege to work with a team of very brilliant people that guided me for the past four years. Starting with my promotors Jeroen van der Laak and Nico Karssemeijer from whom I have learned everything I know about science and academia. I would like to give special thanks to them for trusting me through these years, keeping an open mind, and providing the freedom, independence and opportunities necessary to conduct my research. Moreover, I am also grateful to my daily supervisors, Francesco Ciompi and Geert Litjens, for being such an excellent role model on scientific integrity, hard work, and career success. I deeply appreciate the constant support and confidence that you have placed on me during these years, but also your patience and care, your advice, and your mentorship. Last but not least, I would like to extend my gratitude to all my paper co-authors without whom it would not have been possible to publish this thesis.

Joining the Computational Pathology Group in Nijmegen in 2016 was probably one of the best decisions that I have ever made. We started with a small group of about five PhD students. I was coming from another job in Budapest, Hungary, and I discovered, quite surprisingly, that my next desk partner in Nijmegen was also Hungarian, Péter. We have been through all kinds of moments together: happy, sad, late working hours (very late), but also lots of parties (egészségedre!). Those early moments were quite intense, with people like Babak (probably the best adjective for him is ”legendary”), Oscar (who can always recommend a good pick for mountaineering equipment or a Rammstein song), and Maschenka (the only person that *actually* knows something about pathology from all of us).

The group has greatly expanded since then with extraordinary people, however, not everyone is new in the building. An old story tells that Meyke has been working in the department of pathology since the hospital was built many years ago. Special thanks to her for teaching us, the computer science nerds, how to behave in front of girls and fit in society (pretend to at least). There is also my good old friend Wouter, with whom I have been in the most varied situations: from sharing a bed in multiple occasions, to walking through a space shuttle, and visiting a shooting range in Texas. I cannot forget of Thomas, my bouldering master, who showed me that it is possible to do a fingertip pull up (that it is possible for him, not for the rest of us). Daan, I love your stories and anecdotes, please pause your PhD and write them in a book. Dear reader if you need to buy a second-hand car contact John-Melle, he is an expert and would be delighted to help you (for a small commission). Jasper, Tony Hawk did his famous 900 trick when he was 31, so you can still be a skateboard star. During these years, many more people have helped me in one way or another: Mart,

Zaneta, Caner, Witali, Hans, Esther, Merijn, Gabriel, Irene... thank you all!

I would like to acknowledge many more people from the Diagnostic Image Analysis Group and the department of radiology. I had the pleasure to attend four DIAG weekends during these years, and I can say without a doubt that these were some of the most fun days of my entire PhD. I enjoyed these long nights of karaoke and beer pong, boardgames, playing twister with Clarisa and Nico, tasting handmade pizzas from Francesco, narrated by the best host I know, Midas, and competing for legendary prizes in the DIAG pub quiz. None of these past years would have been the same without Bart, Thomas van den Heuvel, Kevin, Anton, Colin, Bram, Henkjan, Rashindra, Jonas, Ajay, Ecem, Jonas, Paul, Sil, Christiana, Gabriel, Arnaud, Carl, Leti, Suzan, Kaman, Miguel, Marta, Joana, Marco, Fokko, and many many more. A special mention to the bouldering team (Erdi, Luuk, Tariq, Riccardo and Carlijn), and the skiing team (Corina, Felicitas)!

Nijmegen is such an amazing city that has given me many opportunities not only to advance in my science career but also to enjoy life and make great friendships. Cristina, professional drama queen and karaoke partner, we have both learned so much from each other, and have enjoyed many nights out together. Alejandro, the first person I met in Nijmegen, a true friend always there (particularly for those long nights of *sushi and sashimi*) and long term PUBG partner. Special mention to Andres and Belen, my favorite *teleco* friends, who brought me countless fun evenings, both physically and virtually. Ana and Julia, Julia and Ana, difficult to distinguish who is the angel and who the devil, but always ready to rock, hiit, get lost in Vierdaagse, and go to a Macumba party. Big thanks to all my volleyball friends for bringing fun and joy regardless of the weather outside: Jordi, Dimi, Jacopo, Tiz, Fleur, Dani, Santi, Miro, and many more.

Last but not least, I would like to thank my family and friends in Spain, whose unconditional support has been fundamental to me during these past years. I am extremely proud of Tere, my mother, who works as a nurse in my hometown's hospital for having fought bravely during the first wave of the COVID-19 pandemic. She has put her health at risk and has worked tirelessly around the clock to help others in need. For this and many other reasons, she is and will ever be my hero and inspiration. Thanks also to Jose, my father, for his guidance and advice in good and bad moments, and for helping me to understand Van Gogh. Moreover, thanks to my brother and sisters for being a constant source of happiness and fun (Miriam, Manuel and Laura). Finally, a big thank you to my friends in Spain for having sup-

ported me for all these years: Espe, Antonio, Julia, Marta, Manuel, Lola, Alex, Strong Fernando, Fernando Martin... thank you all!

Finalmente, y no menos importante, me gustaría agradecer a mi familia y amigos en España por su apoyo incondicional, el cual ha sido fundamental para mí durante estos últimos años. Me siento muy orgulloso de Tere, mi madre, que trabaja como enfermera en el hospital de mi ciudad natal, por haber luchado con valentía durante la primera ola de la pandemia del COVID-19. Por haber puesto su propia salud en riesgo y haber trabajado sin descanso durante semanas para ayudar a aquellos que lo necesitaban. Por todo ello y muchas más razones, ella es y siempre será mi heroína y mi inspiración. Muchas gracias también a José, mi padre, por sus consejos en los buenos y en los malos momentos, y por haberme ayudado a entender a Van Gogh. También, gracias a mi hermano y hermanas, por ser una fuente constante de felicidad y diversión (Miriam, Manuel y Laura). Finalmente, muchísimas gracias a mis amigos en España por haberme apoyado todos estos años: Espe, Antonio, Julia, Marta, Manuel, Lola, Alex, Fernando Petado, Fernando Martin... gracias a todos!

Curriculum Vitae

David Tellez was born in Cordoba, Spain, in January of 1991. After finishing high school, he graduated in Telecommunication Engineering at the University of Seville in 2014. During his Master's studies he was involved in the Robotics, Vision and Control Research Group as a Research Intern. His Master's Thesis entitled "EEG Signal Processing in Neuromarketing Research" was written involving two international partners: the Budapest University of Technology and Economics, and Synetiq Ltd.

After graduation, he joined Synetiq Ltd in Budapest, Hungary, as the first Research Engineer to develop and scale Synetiq's ambitious vision. For almost two years, he worked there designing and implementing solutions to model and analyze multimodal physiological data, including signals like EEG (brain), EDA (skin), PPG (pupil) and eye-tracking. During these years, the startup evolved from zero revenue and a pilot study, to several recurrent customers and a portfolio of paid products and services.

In February of 2016, David joined the Computational Pathology and Diagnostic Image Analysis Group under the supervision of Dr. Francesco Ciompi and Dr. Jeroen van der Laak as a PhD student. He was a recipient of the Junior Research Grant from Radboud Institute for Health Sciences in 2016 with the goal of developing deep learning based algorithms to improve histopathology image analysis and patient prognostication within Computational Pathology.

His research about deep learning based methods in Computational Pathology has been published in prestigious journals such as the IEEE Transactions on Medical Imaging, the Elsevier Medical Image Analysis, and the IEEE Transactions on Pattern Analysis and Machine Intelligence. His research interests continue to focus on improving healthcare using artificial intelligence and delivering value to patients.

PhD Portfolio

Name: David Tellez Martin
Graduate school: Radboud Institute for Health Sciences (RIHS)
PhD period: 01-02-2016 until 31-01-2020

Courses & workshops	Year(s)	ECTS
Introduction day Radboudumc	2016	0.25
RIHS introduction course for PhD students	2016	1
International Computer Vision Summer School	2016	5
Dutch language course	2016	5
Scientific writing for PhD candidates	2017	3
NFBIA Summer School	2017	3
Deep Learning Specialization on Coursera	2018	5
Internship at Montreal Institute for Learning Algorithms	2019	5
Grant writing and presenting for funding committees course	2019	2

Seminars & lectures		
Radboud research rounds	2016	0.25
Women's cancer meetings	2016	0.25
Scientific lunch meeting of the Pathology department	2016-2020	3
Breast cancer research meetings of the Radiology department	2016	0.25
Deep learning Nijmegen Meetup	2018	0.5
DIAG Discussion Hour	2016-2020	5
Deep learning journal club	2017-2020	4
Computational pathology meeting	2016-2020	5

Symposia & congresses		
Radboud Frontiers	2016	0.25
RIHS PhD retreat	2016-2020	3
International Symposium on Biomedical Imaging	2016	2
Medical Image Computing and Computer Assisted Intervention	2016	2
European Conference on Computer Vision	2016	1
European Conference of Pathology	2016-2017	1
SPIE medical imaging	2018	2
Medical imaging with deep learning	2018, 2020	2
Program committee of COMPAY workshop	2018	1

Teaching & supervision		
Supervision of a Master student	2017	1
Teaching assistant at Intelligent Systems in Medical Imaging	2017-2018	1.5

Research Data Management

The research project described in this PhD thesis makes use of an extensive amount of data with the purpose of training and evaluation several machine learning algorithms. This data consists of three main components: (1) digitized whole-slide images of patient tissue, (2) pixel-level annotations associated to these images, and (3) labels that describe these images at patient and slide level.

Regarding the origin, ownership, and permission to use this data, we strictly follow the regulations of the Radboudumc. For each of the datasets used in this research, we have entered data license agreements with the datasets' stakeholders and have obtained permission to use the data for research purposes.

All this data is securely stored within the Radboudumc storage system. More generally, all scientific experiments conducted within the context of this research project have been executed exclusively within the Radboudumc IT infrastructure.

In order to protect patients' privacy rights, all data used within the context of this research project has been subject to pseudonymization. This process ensures that personally identifiable information is replaced by artificial identifiers, or pseudonyms, before conducting any of the experiments described within this thesis. We adhere to the FAIR data principles (findable, accessible, interoperable and re-usable) whenever possible.